LA-UR-05-7571

Approved for public release; distribution is unlimited.

Title:

Moment-of-fluid interface reconstruction

Author(s): Vadim Dyadechko vdyadechko@lanl.gov

Mikhail Shashkov shashkov@lanl.gov

Mathematical Modeling and Analysis Group, T-7

Los Alamos National Laboratoty

Submitted to: http://math.lanl.gov/~vdyadechko/doc/2005-mof.pdf

Moment-of-fluid interface reconstruction

Vadim Dyadechko[*][†] Mikhail Shashkov[†]

Draft of January 25, 2006

Abstract

We designed a new mass-conservative interface reconstruction method. Unlike volume-of-fluid (VoF) methods, which calculate the interface location from the *volumes* of the cell fractions occupied by different materials, the new algorithm localizes the interface based on both *volumes and centroids* of the cell fractions.

The amount of information carried by the volume and centroid is sufficient to bound the respective cell fraction with a linear interface (i.e. perform Piecewise-Linear Interface Calculation (PLIC)) even *without using the data from the adjacent cells*. The normal of the linear interface can be determined by fitting the centroid of the fraction behind the interface to the given centroid, subject its volume matches the given one.

This technique reconstructs linear interfaces exactly, yields a second order accurate approximation to the interfaces given by simple C^2 curves, has higher resolution and is shown to be more accurate than VoF-PLIC methods.

1 Introduction

There is a variety of discrete interface models developed for Eulerian simulations of a multi-phase fluid flow:

- *Level sets* methods [26, 25, 27, 18] work with *implicit* representation of the interface as a zero level set of a *discrete signed distance function* defined for each pair of adjacent fluid phases.

[*] corresponding author; e-mail: `vdyadechko@lanl.gov`

[†] Mathematical Modeling and Analysis Group (T-7) at Los Alamos National Laboratory

- *Front tracking* methods [30, 8, 23, 29] use supplementary Lagrangian surface grid that marks the interface between different phases.
- *Interface capturing* methods calculate the interface location at each discrete moment of time from the solution data. Two major groups of interface capturing methods should be mentioned here: *volume-of-fluid (VoF)* [6, 15, 9, 31, 4] methods, which construct the interface from the *volume*s of the cell *fractions* occupied by different fluid phases, and *Lagrangian particle* methods [11, 7], which separate the clouds of the test particles of different origin.

Either approach has its virtues and drawbacks. The resolution of level set and VoF methods, for example, is limited by the resolution of the grid, while front tracking and particle methods do not have this limitation. Particle methods are extremely flexible but very expensive, and therefore are feasible mostly as a supplement to other major technique. Front tracking, on the other side, is much cheaper, but only as long as the interface topology is not evolving: the modification the surface grid topology is a challenging task. Neither of level set, VoF, or particle methods experience any problem adopting topological changes of the interface.

What really distinguishes VoF methods from all other techniques is their ability to preserves the mass of each fluid component[1]. For a wide range of applications the importance of *mass conservation* on the discrete level outbalances all potential disadvantages of VoF technology.

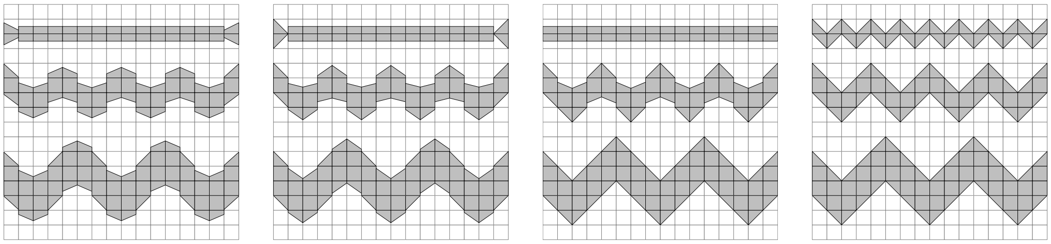
A typical simulation cycle of VoF method consists of two step:

- the advection and
- the interface reconstruction.

One may consider the interface reconstruction to be just an auxiliary for the advection, yielding more accurate calculation of the fluxes. But since *the primary object of our discussion is the interface reconstruction*, we prefer to put the reconstruction algorithm at the center of the picture and leave the advection scheme to provide it with the input data. Moreover, following the divide-and-conquer principle, we are going to take the interface reconstruction completely out of the dynamic VoF context and study it separately. The input date (volume fractions) in this case come not from the advection scheme but from some static multimaterial[1] pattern, and the objective of the algorithm is to approximate the original pattern as accurate as possible.

All VoF interface reconstruction algorithms are designed to separate *two materials*. Volume fractions of two fluid components sharing a mixed cell are not independent but complement each other to 1. Essentially only the volume fractions

[1] Terms “fluid component” and “material” are used here as synonyms of “phase”.



LSG *LVIRA* *Swartz* *MoF = original*

Figure 1. Interface approximations obtained with different PLIC algorithms. MoF method is able to reproduce the original shape exactly.

of a single component, further referred to as a *reference* one, explicitly participate in the interface reconstruction.

The most common interface approximation used by VoF methods consists of a single linear interface per mixed cell (Piecewise-Linear Interface Calculation (PLIC)). Once the direction of the interface outward (with respect to the reference component) normal is know, the location of the interface is uniquely identified by the volume of the reference fraction. There is a number of ways to define the direction of the normal:

- by estimating the gradient of the *discrete volume fraction function* [31, 2];
- by finding a common linear interface for the cluster of cells that guarantees the best approximation to the given volume fractions [20, 19];
- by averaging the normals of the common linear interfaces due to all mixed neighbours [6, 28].

All techniques based on the volume fraction gradient estimation, like Youngs [31], Green-Gauss [2], and Least Square Gradient (LSG) [2] algorithms, allow *direct* implementation and are first order accurate. To get a second order accurate approximation on an unstructured grid, one has to use an *iterative technique*, like Least square VoF Interface Reconstruction Algorithm (LVIRA) [20, 19] or (Mosso-)Swartz algorithm [28, 12].

Regardless of the algorithm employed, the evaluation of the interface normal in a mixed cell requires the volume fraction data from the surrounding cells. As a result, the discrete interfaces in the adjacent mixed cells are never quite independent. This inherent feature of VoF methods prohibits the resulting approximation to resolve any interface details smaller than a characteristic size of the cell cluster involved in the evaluation of the normal; the statement is illustrated by Figure 1, which demonstrates how the quality of LSG, LVIRA, or Swartz interface reconstruction varies with the scale of details in the original shape.

In order to overcome this limitation of VoF methods we propose to enrich an interface reconstruction input data set with the *centroids of the cell fractions.* The amount of information carried by the volumes and centroids or, equivalently, by the *first two moments* of the cell fractions is sufficient to define a *mass-conservative* PLIC approximation *even without sharing the data between the cells*. The normal of the interface in each mixed cell is is determined by *fitting the centroid of the cell fraction behind the interface to the reference centroid, subject its volume matches the reference one*. This strategy, called *moment-of-fluid (MoF) interface reconstruction*, results in second order accurate approximation for C^2-smooth original interfaces; linear interfaces are reconstructed *exactly* (Figure 1, MoF).

With no data from adjacent cells participating in the evaluation of the interface, the method is able to resolve interface details as small as the cell itself, i.e. two to three times smaller than conventional VoF-PLIC methods (see Figure 1). No data exchange also means that *the grid structure is irrelevant for MoF interface reconstruction*.

Compared to alternative approaches, which exploit purely geometrical principles, centroid data involvement has a clear mechanical reason. An *inexact* interface reconstruction introduces some redistribution of the fluid inside a mixed cell. This fluid motion is unrelated to any physical force presented in a discrete model. Any displacement $\Delta\mathbf{x}$ of the cell fraction centroid caused by the interface reconstruction can be interpreted as an action of an *artificial* force of magnitude $\sim m\Delta\mathbf{x}/\Delta t^2$ (here m is the mass of the cell fraction, and Δt is the time increment). Therefore by complying with original centroids we *explicitly* reduce these artificial forces and improve approximation properties of the discrete model of fluid dynamics.

The idea to employ the centroid data in VoF contex has been around for a while. J. Saltzman [21] evaluated the interface normal as the difference between the center of the mixed cell and the centroid of the reference component enclosed in the cluster formed by the cell and its direct neighbours; the centroid estimate obtained from the respective volume fractions yields a first order accurate [19] interface approximation. S. Mosso and S. Clancy [13] used centroid information to derive the material order for multi-material interface reconstruction. The same technique was independently presented by D. Benson [3]. Unfortunately all these initiatives did not get any further development.

The body of the paper is organized into three sections. Section 2 contains the statement of the moment-of-fluid interface reconstruction problem for a single mixed cell, discussion of the solution properties, and the detailed description of MoF algorithm in 2D. Section 3 compares MoF algorithm against VoF counterparts by means of static tests. And Section 4 gives an example of the advection scheme that can be used with MoF interface reconstruction algorithm for simulation of the incomplessible fluid flow.

2 Moment-of-fluid interface reconstruction

Let us start with formal definitions. Consider a mixed cell given by an *open polygon* $\Omega \subset \mathbb{R}^2$. Each cell fraction occupied by a single component may be represented by a *non-trivial open subset* $\omega \subset \Omega$ $(\omega \neq \emptyset,\ \omega \neq \Omega)$. The *family of all non-trivial open subsets of* Ω we denote as $\mathfrak{S}_\Omega$. As a non-empty open set, $\omega \in \mathfrak{S}_\Omega$ always has a positive *volume (area in 2D)*

$$|\omega| \equiv \int\limits_{\omega} \mathrm{d}\mathbf{x}, \qquad 0 < |\omega| < |\Omega|$$

and a well defined *centroid*

$$\mathbf{x}_c(\omega) \equiv \frac{1}{|\omega|} \int\limits_{\omega} \mathbf{x}\,\mathrm{d}\mathbf{x}, \qquad \mathbf{x}_c(\omega) \in \mathrm{hull}(\Omega);$$

here $\mathrm{hull}(\Omega)$ is the *convex hull* of Ω.

In practice it is common to specify the volume of $\omega \in \mathfrak{S}_\Omega$ in terms of *volume fraction*:

$$\mu(\omega) = \frac{|\omega|}{|\Omega|}, \qquad 0 < \mu(\omega) < 1.$$

The part of the subset boundary $\partial\omega$ different from the cell boundary $\partial\Omega$ represents the *interface* $\Gamma(\omega)$ between ω and its complement $\Omega \setminus \overline{\omega}$:

$$\Gamma(\omega) = \partial\omega \setminus \partial\Omega.$$

Whenever the interface is *linear*, the subset is called *truncation volume* or *cutout*. Formally, each cutout $\omega_h \in \mathfrak{S}_\Omega^h$ is a *non-trivial* intersection

$$\omega_h \equiv \Omega \cap h$$

of Ω with an *open half-plane* $h \subset \mathbb{R}^2$; this should also explain the "h" subscript in cutout notation. The *family of all cutouts* we denote as $\mathfrak{S}_\Omega^h$.

2.1 Problem formulation

Suppose Ω contains only two fluid components, and $\omega^* \in \mathfrak{S}_\Omega$ is the original fraction occupied by the reference component. Given the first two moments of ω^*

or, equivalently, the *reference volume fraction* $\mu^* \equiv \mu(\omega^*)$ and the *reference centroid* $\mathbf{x}^* \equiv \mathbf{x}_c(\omega^*)$, one is required to find a cutout $\omega_h^* \in \mathfrak{S}_\Omega^h$ that approximates the original fraction ω^*.

The first natural step would be to find $\omega_h^* \in \mathfrak{S}_\Omega^h$ that matches both given reference moments *exactly*. Unfortunately such a strategy is not guaranteed to succeed. The half-plane h that defines a truncation volume is uniquely identified with 2 independent parameters: the polar angle of the outward unit normal on ∂h and the distance of ∂h from the origin. It means that the *search space* $\mathfrak{S}_\Omega^h$ *is two-dimensional.* The *reference data space*, formed by reference volume fraction and 2 components of reference centroid, *is three-dimensional.* As one can see, it is not always possible to find a cutout that matches given reference moments exactly: in general, such a problem is *overdetermined*. More detailed analysis reveals an important

Property 1. *Each truncation volume is uniquely identified by its centroid.*

Therefore, once the reference centroid $\mathbf{x}^*$ is matched by $\mathbf{x}_c(\omega_h^*)$, there is no extra degree freedom to match the reference volume. Nevertheless if the original reference fraction ω^* is a truncation volume itself, the cutout ω_h^* with $\mathbf{x}_c(\omega_h^*) = \mathbf{x}^*$ has no choice but to match the original cell fraction ω^* *exactly*; in this case the reference volume fraction is matched automatically.

Since the volume (mass) conservation is of highest priority for us, we have to sacrifice the exact centroid matching in the case when $\omega^* \notin \mathfrak{S}_\Omega^h$. Our strategy is to *find a cutout of the given volume fraction whose centroid provides the best approximation to the reference one.* This leads to a formal optimization problem in the *space of cutouts of the given volume fraction*:

$$\mathfrak{S}_\Omega^{h,\mu^*} \equiv \{\ \omega_h \in \mathfrak{S}_\Omega^h \mid \ \mu(\omega_h) = \mu^*\ \}.$$

Problem 1. *Find a cutout* $\omega_h^* \in \mathfrak{S}_\Omega^{h,\mu^*}$ *such that*

$$\omega_h^* = \arg \min_{\omega_h \in \mathfrak{S}_\Omega^{h,\mu^*}} ||\,\mathbf{x}_c(\omega_h) - \mathbf{x}^*||^2, \tag{1}$$

where $||\cdot||$ *is the Euclidean norm.*

This problem, further referred to as a *moment-of-fluid (MoF) interface reconstruction*, makes the object of our study.

2.2 Mathematics of MoF interface reconstruction

Now we would like to give an insight into Problem 1. The reader will find here no formal proofs, only declarations of important facts along with some illustrations. The material presented in this subsection is not essential for understanding the rest of the paper and can be skipped with no harm. Still, we would like the impatient reader to be aware of the main results reported: *MoF interface reconstruction is unique (with absolute certainty), stable (also with absolute certainty), and second order accurate (for sufficientlty smooth interfaces).*

Let us (re-)introduce the following families of cell fractions:

- all cutouts of the given volume fraction:

$$\mathfrak{S}_\Omega^{h,\mu^*} = \{\, \omega_h \in \mathfrak{S}_\Omega^h \mid \mu(\omega_h) = \mu^* \,\};$$

- all non-trivial open subsets of the given volume fraction:

$$\mathfrak{S}_\Omega^{\mu^*} = \{\, \omega \in \mathfrak{S}_\Omega \mid \mu(\omega) = \mu^* \,\};$$

and the loci of respective centroids:

$$\begin{aligned} \mathcal{X}_\Omega^{h,\mu^*} &\equiv \mathbf{x}_c(\mathfrak{S}_\Omega^{h,\mu^*}) = \{\ \mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = \mathbf{x}_c(\omega_h),\ \omega_h \in \mathfrak{S}_\Omega^{h,\mu^*}\ \}, \\ \mathcal{X}_\Omega^{\mu^*} &\equiv \mathbf{x}_c(\mathfrak{S}_\Omega^{\mu^*}) \ \ = \{\ \mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = \mathbf{x}_c(\omega_h),\ \omega \in \mathfrak{S}_\Omega^{\mu^*}\ \}. \end{aligned}$$

Since each truncation volume is uniquely identified by its centroid (Property 1), Problem 1 is *equivalent* to finding a nearest to $\mathbf{x}^*$ point of locus $\mathcal{X}_\Omega^{h,\mu^*}$:

Problem 1a. *Find a point* $\mathbf{x}_h^* \in \mathcal{X}_\Omega^{h,\mu^*}$ *such that*

$$\mathbf{x}_h^* = \arg \min_{\mathbf{x}_h \in \mathcal{X}_\Omega^{h,\mu^*}} ||\,\mathbf{x}_h - \mathbf{x}^*||^2. \tag{1a}$$

Although this alternative formulation of MoF interface reconstruction problem is of little practical use, it provides a simple geometrical model to analyze the *existence*, *uniqueness*, and *stability* of MoF interface reconstruction.

Since Problem 1a is about approximating $\mathbf{x}^* \in \mathcal{X}_\Omega^{\mu^*}$ with elements of $\mathcal{X}_\Omega^{h,\mu^*}$, our attention should be focused on the properties of these two loci:

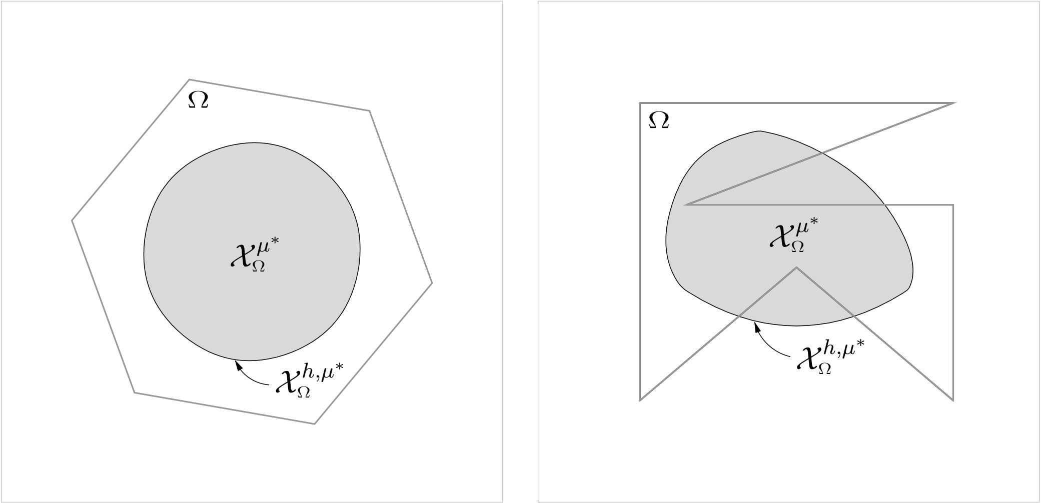
Property 2. $\mathcal{X}_\Omega^{\mu^*}$ *is a strictly convex closed set with a smooth boundary*

$$\partial\mathcal{X}_\Omega^{\mu^*} = \mathcal{X}_\Omega^{h,\mu^*}.$$

This property is illustrated by Figure 2.

Property 3. *Among all the elements of* $\mathfrak{S}_\Omega^{\mu^*}$, *only cutouts have their centroids located on the boundary of* $\mathcal{X}_\Omega^{\mu^*}$.

Yet another property, even though it is not referenced directly, completes the picture of the problem we study:


Ω
$\mathcal{X}_\Omega^{\mu^*}$
$\mathcal{X}_\Omega^{h,\mu^*}$
Ω
$\mathcal{X}_\Omega^{\mu^*}$
$\mathcal{X}_\Omega^{h,\mu^*}$

convex cell *non-convex cell*

Figure 2. Examples of $\mathcal{X}_\Omega^{\mu^*}$ and $\mathcal{X}_\Omega^{h,\mu^*}$ loci ($\mu^* = \mathit{0.25}$) for polygonal cells of different shape.

Property 4. *Centroid loci given by different values of volume fraction compose a uniparametric family*

$$\{\ \mathcal{X}_\Omega^{\mu^*} \mid 0 < \mu^* < 1\ \}$$

of nested sets:

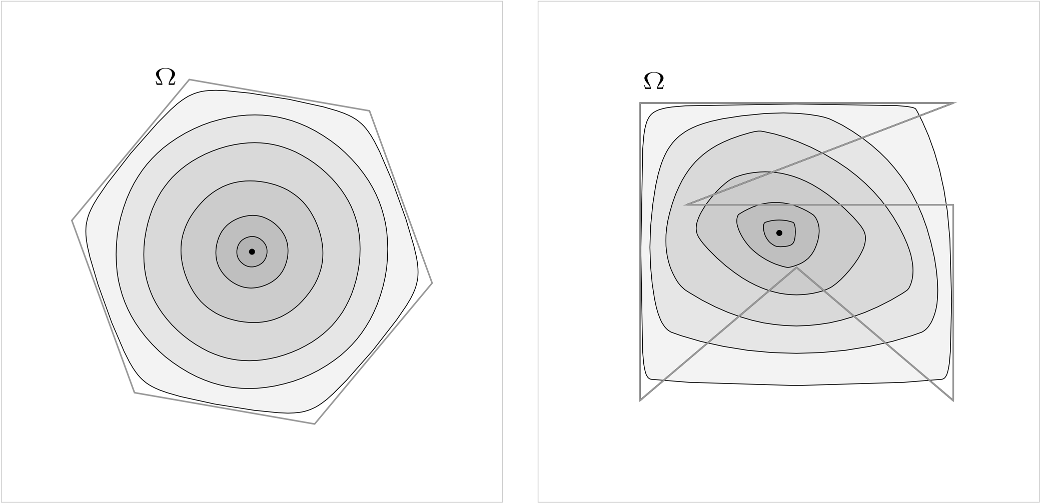
$$\begin{aligned} \mu_1^* < \mu_2^* \quad &\Longrightarrow \quad \mathcal{X}_\Omega^{\mu_2^*} \subset \mathcal{X}_\Omega^{\mu_1^*}, \\ \bigcap_{0<\ \mu^*<1} \mathcal{X}_\Omega^{\mu^*} \quad &= \quad \mathbf{x}_c(\Omega), \\ \bigcup_{0<\ \mu^*<1} \mathcal{X}_\Omega^{\mu^*} \quad &= \quad \operatorname{hull}(\Omega). \end{aligned}$$

Figure 3 shows examples of nested centroid loci defined by different values of the reference volume fraction.

Existence. The search space ($\mathcal{X}_\Omega^{h,\mu^*}$) of continuous objective function $||\,\mathbf{x}_h - \mathbf{x}^*||^2$ is compact (Property 2), therefore by Weierstrass theorem solution of the minimization problem (1a) *always exists*.

Uniqueness. Even though reference moments are assumed to be given by some cell fraction, i.e. $\mathbf{x}^* \in \mathcal{X}_\Omega^{\mu^*}$, one better not count on this fact in a real life calculation: miscellaneous discretization and round-off errors may cause the input data to be inconsistent.

Properties 2 and 3 allow us to classify all input data $(\mu^*, \mathbf{x}^*) \in\ ]\,0, 1\,[\times\mathbb{R}^2$ into three categories (see Figure 4a):



convex cell

non-convex cell

Figure 3. Illustration of the nested structure of $\{\ \mathcal{X}_\Omega^{\mu^*} \mid 0 < \mu^* < 1\ \}$ locus family for polygonal cells of different shape: $\mathbf{x}_c(\Omega) \in \mathcal{X}_\Omega^{0.90} \subset \mathcal{X}_\Omega^{0.75} \subset \mathcal{X}_\Omega^{0.50} \subset \mathcal{X}_\Omega^{0.25} \subset \mathcal{X}_\Omega^{0.10} \subset \mathcal{X}_\Omega^{0.05} \subset \mathrm{hull}(\Omega)$.

1) $\mathbf{x}^* \notin \mathcal{X}_\Omega^{\mu^*}$: the reference moment data are inconsistent, i.e. there is no cell fraction that satisfies them. Some consolation is supplied by the fact that solution of (1) (which is *unique* because $\mathbf{x}^*$ is located outside the convex region bounded by $\mathcal{X}_\Omega^{h,\mu^*} \equiv \partial\mathcal{X}_\Omega^{\mu^*}$) gives *the best* approximation to the reference centroid *among all the subsets of the given volume fraction* μ^*.

2) $\mathbf{x}^* \in \partial\mathcal{X}_\Omega^{\mu^*}$: the source of the reference data is known to be a truncation volume (Property 3), which can be *uniquely* (Property 1) identified by solving (1).

3) $\mathbf{x}^* \in \mathcal{X}_\Omega^{\mu^*} \setminus \partial\mathcal{X}_\Omega^{\mu^*}$: the original cell fraction ω^* is *not* a cutout. This type of the input is the most common and also requires more careful analysis.

   To start with, Problem 1a in this case may have multiple solutions (Figure 4b). The number of multiple solutions is always finite and can be as big as the number of vertices of the polygon given by the convex hull of a cell.

   We use symbol $\tilde{\mathcal{X}}_\Omega^{\mu^*}$ to denote the *locus of all reference centroids defining multiple solutions of (1a)*. $\tilde{\mathcal{X}}_\Omega^{\mu^*}$ is a non-empty connected set composed of a finite number of smooth segments that form a tree graph pattern (see Figure 4b); the tips of the tree branches are always open. Since $\tilde{\mathcal{X}}_\Omega^{\mu^*}$ has zero area, one can consider the input $\mathbf{x}^* \in \tilde{\mathcal{X}}_\Omega^{\mu^*}$ to be *improbable*. Therefore *MoF interface*

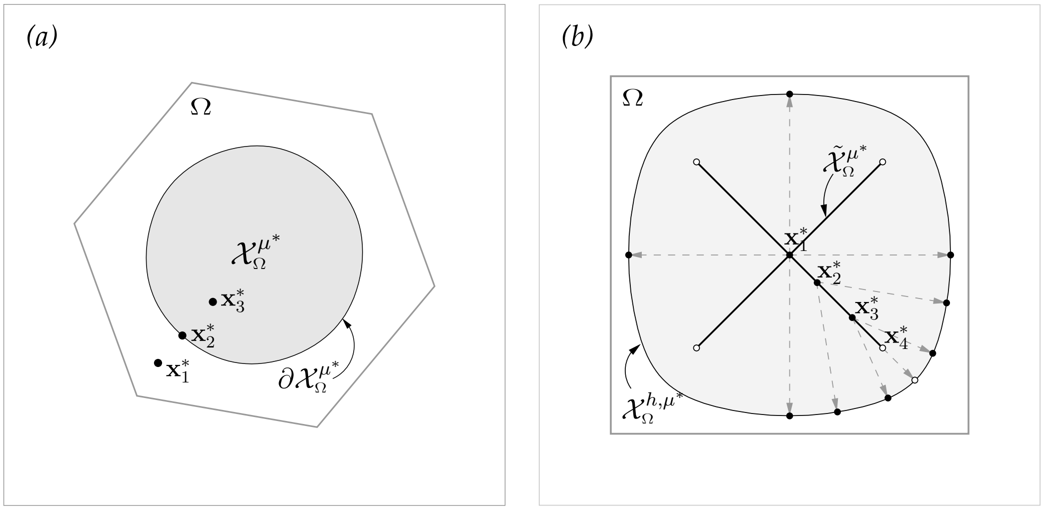

(a)
Ω
$\mathcal{X}_\Omega^{\mu^*}$
$\mathbf{x}_3^*$
$\mathbf{x}_2^*$
$\mathbf{x}_1^*$
$\partial\mathcal{X}_\Omega^{\mu^*}$
(b)
Ω
$\tilde{\mathcal{X}}_\Omega^{\mu^*}$
$\mathbf{x}_1^*$
$\mathbf{x}_2^*$
$\mathbf{x}_3^*$
$\mathbf{x}_4^*$
$\mathcal{X}_\Omega^{h,\mu^*}$

Figure 4. (a) For a given value of μ^* the closed curve $\mathcal{X}_\Omega^{h,\mu^*} \equiv \partial\mathcal{X}_\Omega^{\mu^*}$ divides all possible locations of reference centroid into three groups: strictly outside $\mathcal{X}_\Omega^{h,\mu^*}$ ($\mathbf{x}_1^*$ sample), directly on $\mathcal{X}_\Omega^{h,\mu^*}$ ($\mathbf{x}_2^*$ sample), and strictly inside $\mathcal{X}_\Omega^{h,\mu^*}$ ($\mathbf{x}_3^*$ sample).

(b) Example of $\tilde{\mathcal{X}}_\Omega^{\mu^*}$ locus of reference centroids that specifies non-unique solutions of MoF interface reconstruction problem for a square cell and $\mu^* = 0.1$. One can see here reference centroid samples $\mathbf{x}_1^*, \mathbf{x}_2^*, \mathbf{x}_3^* \in \tilde{\mathcal{X}}_\Omega^{\mu^*}$, $\mathbf{x}_4^* \notin \tilde{\mathcal{X}}_\Omega^{\mu^*}$ and respective solutions of Problem 1a located on $\mathcal{X}_\Omega^{h,\mu^*}$. With $\mu^* = 0.1$, there are 4 solutions of the problem for $\mathbf{x}^* = \mathbf{x}_1^*$, 2 solutions for $\mathbf{x}^* = \mathbf{x}_{2,3}^*$, and only one solution for $\mathbf{x}^* = \mathbf{x}_4^*$ (open tip of the $\tilde{\mathcal{X}}_\Omega^{\mu^*}$ brunch).

reconstruction is unique with absolute certainty.

There is one special case worth separate consideration. It can be shown that for each particular shape of a mixed cell Ω, there exists $R > 0$, such that for any cell fraction $\omega^* \in \mathfrak{S}_\Omega$ with the interface curvature uniformly bounded from above by $1/R$, the centroid $\mathbf{x}^* = \mathbf{x}_c(\omega^*)$ is located beyond the reach of $\tilde{\mathcal{X}}_\Omega^{\mu^*}$, $\mu^* = \mu(\omega^*)$. This fact becomes very important when the reference data are *error free*, namely, *if the original interface is given by a C^2-smooth curve with its curvature bounded from above by a positive constant, one can always introduce sufficiently fine shape-regular polygonal mesh to guarantee that MoF interface reconstruction is unique in each mixed cell.*

Stability. MoF interface reconstruction problem depends on a set of input parameters: reference volume fraction and centroid. The solution of a problem for given input parameters is considered to be stable if:

1) it is unique;
2) it stays unique under any sufficiently small perturbation of input parameters;
3) it depends on input parameters continuously.

Each input $(\mu^*, \mathbf{x}^*)$ of MoF interface reconstruction problem can be classified either as *unique* or *multiple*, depending on the number of resulting solutions. If $\mathcal{M}_\Omega$ is a subset of the *input data space* $\mathcal{I}_\Omega \equiv \;]0,1[\times\mathbb{R}^2$ composed of all multiple inputs:

$$\mathcal{M}_\Omega = \{\ (\mu^*, \mathbf{x}^*) \in\ \mathcal{I}_\Omega\ \mid\ \mathbf{x}^* \in \tilde{\mathcal{X}}_\Omega^{\mu^*}\ \},$$

then the complement of $\mathcal{M}_\Omega$:

$$\mathcal{U}_\Omega = \mathcal{I}_\Omega \backslash \mathcal{M}_\Omega = \{\ (\mu^*, \mathbf{x}^*) \in\ \mathcal{I}_\Omega\ \mid\ \mathbf{x}^* \notin \tilde{\mathcal{X}}_\Omega^{\mu^*}\ \}$$

describes all unique inputs.

Not all unique inputs satisfy the second stability criterion: any input that corresponds to the tip of an $\tilde{\mathcal{X}}_\Omega^{\mu^*}$ branch (sample $\mathbf{x}_4^*$ on Figure 4b), while being unique, is a limit point of $\mathcal{M}_\Omega$, i.e. under an infinitesimally small perturbation may turn into a multiple input. By excluding from $\mathcal{U}_\Omega$ all limit points of $\mathcal{M}_\Omega$ we obtain the largest open subset of $\mathcal{U}_\Omega$:

$$\mathcal{S}_\Omega \equiv \mathcal{I}_\Omega \backslash \overline{\mathcal{M}_\Omega} = \{\ (\mu^*, \mathbf{x}^*) \in\ \mathcal{I}_\Omega\ \mid\ \mathbf{x}^* \notin \overline{\tilde{\mathcal{X}}_\Omega^{\mu^*}}\ \},$$

which is composed of all inputs that satisfy the first two stability criteria.

As of the last criterion, it can be proven illustrated by Figure 5a

Property 5. *Solution of MoF interface reconstruction problem continuously depends on reference moments.*

Therefore the set $\mathcal{S}_\Omega \equiv \mathcal{I}_\Omega \backslash \overline{\mathcal{M}_\Omega}$ also describes all stable inputs. Taking into account the fact of $\overline{\mathcal{M}_\Omega}$ being a *measure zero* set in space $\mathcal{I}_\Omega$, one can conclude that *solution of MoF interface reconstruction problem is stable with absolute certainty.*

Approximation properties. There are many possible ways to quantify the interface reconstruction error. One, for instance, can measure either

- the *first moment defect:*

$$\Delta M_1 \equiv ||\,\mathbf{M}_1(\omega_h^*) - \mathbf{M}_1(\omega^*)|| = |\Omega|\,\mu^*||\,\mathbf{x}_h^* - \mathbf{x}^*||, \tag{2}$$

  where $\mathbf{M}_1(\omega) \equiv \int\limits_\omega \mathbf{x}\ \mathrm{dx}$ is the first moment of a cell fraction $\omega \in \mathfrak{S}_\Omega$;

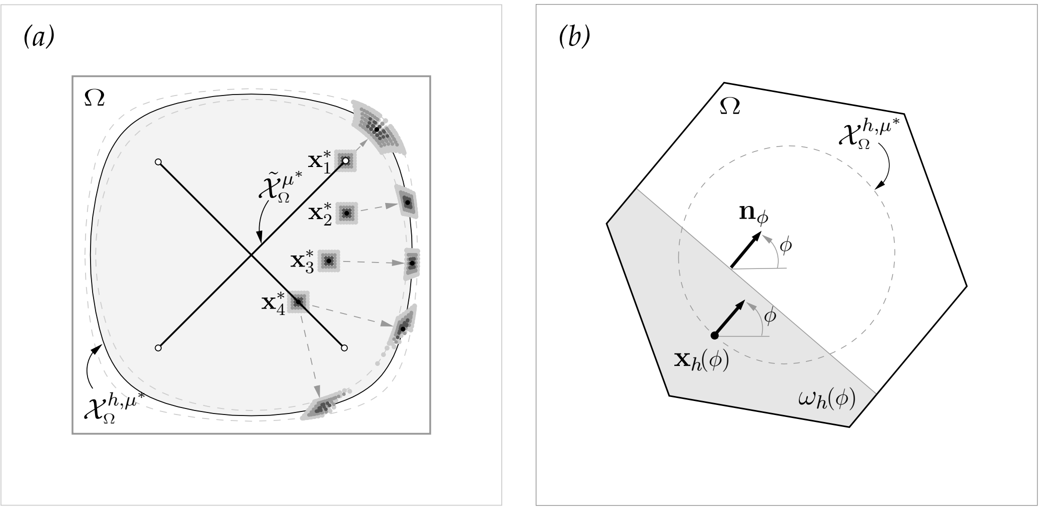


Figure 5. (a) Sample reference centroids $\mathbf{x}_1^*, \mathbf{x}_2^*, \mathbf{x}_3^*, \mathbf{x}_4^*$ and respective solutions diffused by fuzziness of the reference data (of both centroid and volume fraction) demonstrate continuous dependence of MoF solution on reference moments.
(b) The parametrization of the search space for numerical optimization of (1). Each $\omega_h \in \mathfrak{S}_\Omega^{h,\mu^*}$ is uniquely identified by the polar angle ϕ of the interface outward normal $\mathbf{n}$. Note that ϕ also specifies the direction of the local inward normal on $\mathcal{X}_\Omega^{h,\mu^*}$ at point $\mathbf{x}_h(\phi) \equiv \mathbf{x}_c(\omega_h(\phi))$.

- the *area of the symmetric difference* between ω_h^* and ω^*:

$$\Delta\omega \equiv |\,\omega_h^* \triangle\, \omega^*| = |(\,\omega_h^* \cup \omega^*) \setminus (\,\omega_h^* \cap \omega^*)|;$$

- or the *maximum deviation* of the original interface $\Gamma(\omega^*)$ from the cut-line ∂h^* defining $\Gamma(\omega_h^*) = \Omega \cap \partial h^*$:

$$\Delta\Gamma \equiv \text{dist}(\Gamma(\omega^*), \partial h^*) = \max_{\mathbf{x}\in\Gamma(\omega^*)} \min_{\mathbf{y}\in\partial h^*} ||\mathbf{x} - \mathbf{y}||.$$

Since MoF interface reconstruction is achieved by minimizing $||\,\mathbf{x}_h^* - \mathbf{x}^*||$, then according to (2) it should result in the *minimal defect of the first moment attainable with a mass-conservative PLIC approximation*. In this sense *MoF interface reconstruction is optimal in the class of mass-conservative PLIC methods*.

As long as the original interface $\Gamma(\omega^*)$ is known to be a C^2 curve with curvature radius bounded from below by a positive constant R, the following estimates hold true:

$$\Delta M_1 = O(d^5/R^2), \quad \Delta\omega = O(d^3/R), \quad \Delta\Gamma = O(d^2/R), \tag{3}$$

where $d = \operatorname{diam}(\Omega)$. The fact of $\Delta\Gamma$ scaling quadratically with the diameter of the cell d proves that *MoF interface reconstruction is second order accurate*.

For a non-smooth interface all errors above reach their respective pessimistic upper bounds:

$$\Delta M_1 = O(d^3), \quad \Delta\omega = O(d^2), \quad \Delta\Gamma = O(d). \tag{4}$$

2.3 Implementation of MoF strategy in 2D

For a number of simple shapes of Ω MoF interface reconstruction problem can certainly be solved analytically. But our objective is to design a method that is able to treat an arbitrary polygonal mixed cell. That is why we turn to numerical optimization of (1).

Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^2$ be the vertices of a mixed cell Ω enumerated in *counter-clockwise order*. The area and centroid of Ω (as well as of any other polygon with n vertices) can be calculated by formulas:

$$|\Omega| = \frac{1}{2}\sum_{i=1}^{n}[\,\mathbf{x}_i \times \mathbf{x}_{i+1}\,], \tag{5}$$

$$\mathbf{x}_c(\Omega) = \frac{1}{6\,|\Omega|}\sum_{i=1}^{n}[\,\mathbf{x}_i \times \mathbf{x}_{i+1}\,](\mathbf{x}_i + \mathbf{x}_{i+1}), \tag{6}$$

where $[\,\cdot \times \cdot\,]$ is *2D vector product*, and $\mathbf{x}_{n+1} \equiv \mathbf{x}_1$.

In order to proceed with numerical optimization, one have to introduce a suitable parametrization of the search space $\mathfrak{S}_\Omega^{h,\mu^*}$. Each $\omega_h \in \mathfrak{S}_\Omega^{h,\mu^*}$ is uniquely identified with the polar angle ϕ of the interface outward normal $\mathbf{n}(\Gamma(\omega_h))$ (Figure 5b). We consider $\omega_h(\phi) \in \mathfrak{S}_\Omega^{h,\mu^*}$ to be a cutout of the given volume fraction with the interface outward normal $\mathbf{n}(\Gamma(\omega_h(\phi))) = \mathbf{n}_\phi \equiv (\cos\phi, \sin\phi)$, and $\mathbf{x}_h(\phi) \equiv \mathbf{x}_c(\omega_h(\phi))$ to be the respective centroid.

The value of the objective function

$$f(\phi) \equiv ||\,\mathbf{x}_h(\phi) - \mathbf{x}^*||^2 \tag{7}$$

for a given $\phi \in \mathbb{R} \bmod 2\pi$, can be calculated in two steps:

1) find $\omega_h(\phi) \in \mathfrak{S}_\Omega^{h,\mu^*}$;
2) using (6) calculate $\mathbf{x}_h(\phi)$ as the centroid of $\omega_h(\phi)$ and substitute it in (7).

The first step requires a detailed explanation. Instead of finding $\omega_h \in \mathfrak{S}_\Omega^{h,\mu^*}$ with $\mathbf{n}(\Gamma(\omega_h)) = \mathbf{n}_\phi$, it is more convenient to identify $\omega_h(\phi)$ as a cutout of the given volume fraction among the *truncation volumes with prescribed interface normal*

$$\mathfrak{S}_\Omega^{h,\phi} = \{\, \omega_h \in \mathfrak{S}_\Omega^{h} \;\mid\; \mathbf{n}(\Gamma(\omega_h)) = \mathbf{n}_\phi \,\}.$$

Without loss of generality we can assume that the vertical direction is defined by vector $\mathbf{n}_\phi$. This convention allows us to use common terms, like *height* and *altitude*, to specify the size and the position of an object along the direction $\mathbf{n}_\phi$.

Each truncation volume $\omega_{h,\phi} \in \mathfrak{S}_\Omega^{h,\phi}$ is uniquely identified by its height ξ. The cutout volume fraction $\mu(\xi) \equiv \mu(\omega_{h,\phi}(\xi)),\ \omega_{h,\phi}(\xi) \in \mathfrak{S}_\Omega^{h,\phi}$ is a *continuous monotone* function of ξ (Figure 6), i.e. has a well define inverse. Therefore the height ξ^* of $\omega_h(\phi)$ can be found by solving the equation

$$\mu(\xi^*) = \mu^*. \tag{8}$$

The second derivative of $\mu(\xi)$ is a *piecewise-constant* function with discontinuity points given by the vertex altitudes $\xi_1, \xi_2, \ldots, \xi_n$. This fact along with monotonicity of $\mu(\xi)$ yields an efficient solution strategy for (8), which requires only $O(n)$ operations for a convex Ω and $O(n^2)$ operations in general case. We call it *flood algorithm*, since it models the process of flooding a vessel, represented by a cell, with fluid.

Flood algorithm specifies three steps to identify the "level of fluid" ξ^* in the "vessel" Ω, at which the volume fraction below the "surface" $\xi = \xi^*$ matches μ^* :

1) (pre-)sort cell vertices by their altitudes to have all discontinuity points of $\mu''(\xi)$ arranged in a non-descending order;
2) find the interval of quadraticity (linearity) of $\mu(\xi)$ that includes ξ^*;
3) calculate ξ^* by means of polynomial interpolation.

Let $(i_1, i_2, \ldots, i_n)$ be an index permutation that puts vertex altitudes in a non-descending order:

$$0 \equiv \xi_{i_1} \leqslant \xi_{i_2} \leqslant \ \ldots \ \leqslant \xi_{i_n}. \tag{9}$$

First one have to find the interval $[\,\xi_{i_{k^*}}, \xi_{i_{k^*+1}}\,]$ of quadraticity (linearity) of $\mu(\xi)$ that includes ξ^*. Such an interval should satisfy condition

$$\mu_{i_{k^*}} \leqslant \mu^* \leqslant \mu_{i_{k^*+1}}, \quad 1 \leqslant k^* \leqslant n-1, \tag{10}$$

where $\mu_k \equiv \mu(\xi_{i_k})$ is the volume fraction of the cutout of height $\xi_{i_k},\ k = \overline{1,n}$.

Since $\mu(\xi)$ is either quadratic or linear on each interval $[\,\xi_{i_k}, \xi_{i_{k+1}}],\ k = \overline{1, n-1}$, the area of Ω, enclosed between two successive levels ξ_{i_k} and $\xi_{i_{k+1}}$, can be calculated by the *trapezoid rule*:

$$|\Omega|\,(\mu_{k+1} - \mu_k) = \frac{1}{2}\,(\xi_{i_{k+1}} - \xi_{i_k})\,(|\Gamma_{k+1}| + |\Gamma_k|). \tag{11}$$

Here $|\Omega|$ is the area of the cell, and $|\Gamma_k|$ is the total length of the cell cross section $\Gamma_k \equiv \Gamma(\xi_{i_k}) \equiv \Gamma(\omega_h(\xi_{i_k}))$ at level $\xi_{i_k},\ k = \overline{1,n}$.

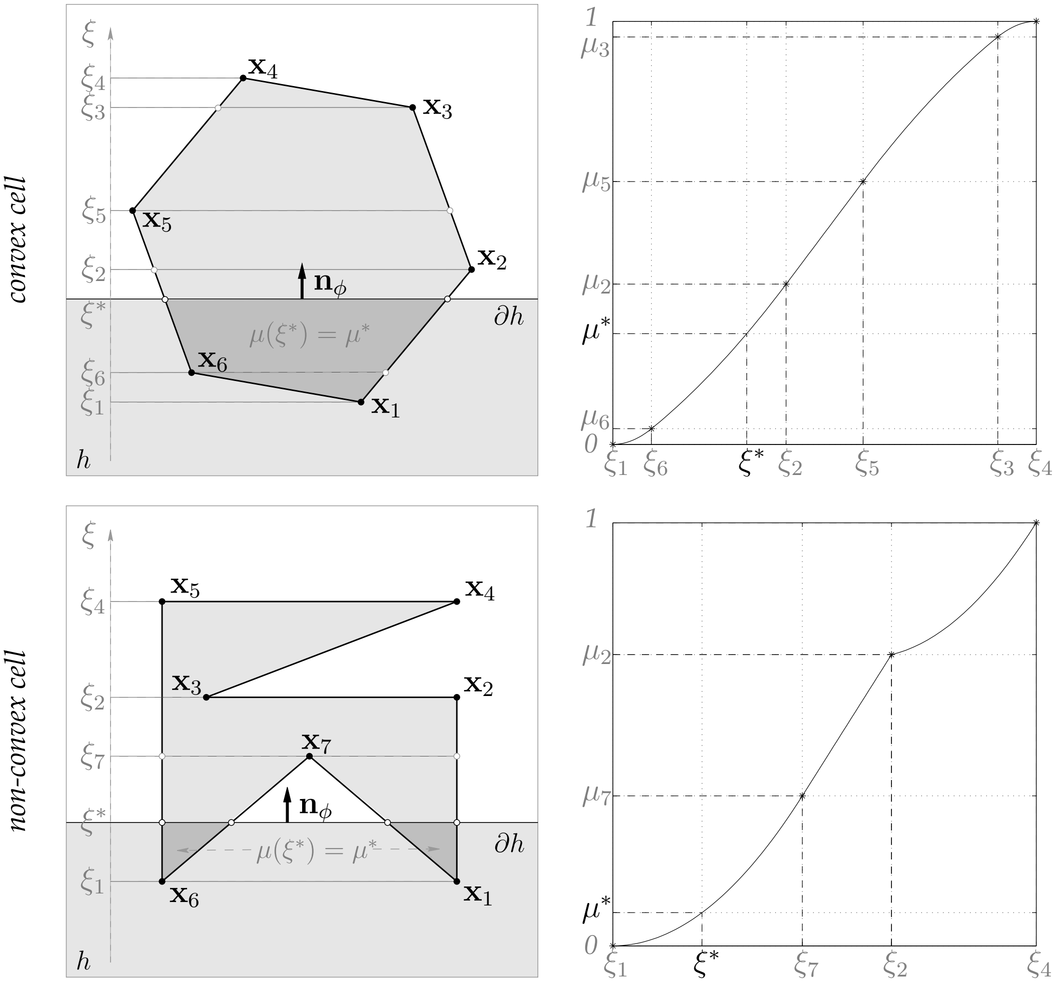


Figure 6. Two different polygonal cells and the respective graphs of the volume fraction $\mu(\xi)$ below the horizontal interface at level ξ.

Expression (11) gives a basis for recurrent calculation of all $\mu_k,\ k = \overline{2, n-1}$. The search starts from the "bottom"

$$\mu_1 = 0, \quad |\Gamma_1| = 0,$$

and goes all the way "up"

$$\mu_k = \mu_{k-1} + \frac{1}{2\,|\Omega|}\,(\xi_{i_k} - \xi_{i_{k-1}})\,(|\Gamma_k| + |\Gamma_{k-1}|), \quad k = 2, 3, \ldots, \tag{12}$$

until the reference volume fraction is bracketed.

Note that each step of iterative process (12) involves identification of respective cross section $\Gamma_k,\ k = \overline{2, n-1}$ ($O(1)$ operations for a convex cell, $O(n)$ otherwise).

Once the interval of interest (10) is known, solution of (8) can be found by means of polynomial interpolation:

- whenever $|\Gamma_{k^*+1}| = |\Gamma_{k^*}|$, the function $\mu(\xi)$ is linear on $[\,\xi_{i_{k^*}}, \xi_{i_{k^*+1}}\,]$, and
$$\xi^* = \xi_{i_{k^*}} + \frac{\mu^* - \mu_{k^*}}{\mu_{k^*+1} - \mu_{k^*}}\,(\xi_{i_{k^*+1}} - \xi_{i_{k^*}});$$
- otherwise $\mu(\xi)$ is quadratic, and
$$\xi^* = \xi_{i_{k^*}} + \frac{|\Gamma^*| - |\Gamma_{k^*}|}{|\Gamma_{k^*+1}| - |\Gamma_{k^*}|}\,(\xi_{i_{k^*+1}} - \xi_{i_{k^*}}),$$
where
$$|\Gamma^*| = \sqrt{|\Gamma_{k^*}|^2 + \frac{\mu^* - \mu_{k^*}}{\mu_{k^*+1} - \mu_{k^*}}\,(|\Gamma_{k^*+1}|^2 - |\Gamma_{k^*}|^2)}$$
is the total length of the interface $\Gamma^* = \Gamma(\omega_h(\phi))$.

Bracketing the reference volume fraction (12) is the most expensive part of the whole algorithm (the complexity of the initial vertex altitude sort (9) is just $O(n)$ for a convex cell and $O(n \log n)$ in general case). One can reduce the expected bracketing time by half by implementing the search (12) in backward order for all $\mu^* > 1/2$. In this case the search starts from the "top" ($\mu_n = 0,\ |\Gamma_n| = 0$) and goes all the way "down" ($k = n-1, n-2, \ldots$) to the interval of interest. Another reason to implement the backward search is more accurate evaluation of ξ^* for $\mu^* \approx 1$.

The derivative of the objective function. By calculating the alteration of the first moment of cutout $\omega_h(\phi)$ due to infinitesimal increment of the interface normal, one can find a simple expression for the first derivative of $\mathbf{x}_h(\phi)$:

- whenever the interface $\Gamma(\phi) \equiv \Gamma(\omega_h(\phi))$ consists of a single segment (which is always the case for a convex cell),
$$\mathbf{x}_h'(\phi) = -\frac{1}{12}\frac{|\Gamma(\phi)|^3}{\mu^*|\Omega|}\,\mathbf{t}(\phi), \tag{13}$$
where $\mathbf{t}(\phi) = (-\sin\phi, \cos\phi)$ is a *counter-clockwise unit tangent* on $\Gamma(\phi)$.
- for a non-convex cell the interface $\Gamma(\phi)$ may consist of several separate segments of the cut-line (see Figure 6, non-convex cell); in this case
$$\mathbf{x}_h'(\phi) = -\frac{M_2(\Gamma(\phi))}{\mu^*|\Omega|}\,\mathbf{t}(\phi), \tag{14}$$

where $M_2(\Gamma(\phi))$ is the *second central moment* of the interface $\Gamma(\phi)$:

$$M_2(\Gamma) = \int\limits_{\Gamma} ||\,\mathbf{x} - \mathbf{x}_c(\Gamma)||^2 \,\mathrm{d}\Gamma, \quad \mathbf{x}_c(\Gamma) = \frac{1}{|\Gamma|} \int\limits_{\Gamma} \mathbf{x}\,\mathrm{d}\Gamma.$$

Expressions (13) and (14) complete the *chain rule* for the first derivative of the objective function:

$$f'(\phi) = 2\,\left((\mathbf{x}_h(\phi) - \mathbf{x}^*) \cdot \mathbf{x}'_h(\phi) \right).$$

Here $(\cdot \cdot \cdot)$ is a *dot product*.

Numerical optimization of the objective function (7) can be performed with any univarite optimization routine available. The iterative procedure we implemented is based on the recepies from [14] and [5]. It takes four input parameters:

1) the *initial guess* ϕ_0,
2) the *trial step increment* $\Delta\phi > 0$,
3) the *argument tolerance* $tol_\phi \geqslant 0$,
4) the *derivative tolerance* $tol_\phi \geqslant 0$;

and returns the iterate ϕ_i considered to be a good approximation to the local minimum ϕ^* of the objective function $f(\phi)$, namely the first ϕ_i that satisfies either of two conditions:

- $|f'(\phi_i)| \leqslant tol_{f'}$ or
- $|\phi_i - \phi^*| \leqslant tol_\phi$.

The prosedure consists of three logical parts:

1) First, the descending direction of $f(\phi)$ is identified.
   If $|f'(\phi_0)| < tol_{f'}$ then the search is over, and ϕ_0 is returned.
   Otherwise if $f'(\phi_0) > 0$, the positive direction of the polar angle is altered from the *counter-clockwise* to the *clockwise* to ensure that $f(\phi)$ is decreasing at $\phi = \phi_0$.
2) Second, the local minimum is bracketed. For this we generate a monotonically increasing *arithmetic progression*

   $$\phi_j = \phi_0 + \Delta\phi \cdot j, \quad j = \overline{1, i},$$

   until either

   - a good approximation to the local minimum is found: $|f'(\phi_i)| < tol_{f'}$, or
   - the objective function growth is detected: $f(\phi_i) \geqslant f(\phi_{i-1})$ or $f'(\phi_i) \geqslant 0$.

If the former condition terminates the loop, then the search is over and ϕ_i is returned.

The latter condition guarantees that the interval $[\phi_{i-1}, \phi_i]$ contains a local minimum. The minimum can be identified with arbitrary accuracy by means of *recursive cubical interpolations* (see the *zoom algorithm* in [14]), which progressively contract the target interval known to contain a local minimum. The end points of the largest target interval are defined as

$$\begin{aligned} \phi_{lo} &\leftarrow \phi_{i-1}, \\ \phi_{hi} &\leftarrow \phi_i \end{aligned}$$

if $f(\phi_i) \geqslant f(\phi_{i-1})$, and

$$\begin{aligned} \phi_{lo} &\leftarrow \phi_i, \\ \phi_{hi} &\leftarrow \phi_{i-1} \end{aligned}$$

otherwise.

3) The order of the end points of the interval ϕ_{lo} and ϕ_{hi} is chosen such that ϕ_{lo}, among all the iterates generated so far, gives the smallest value to the objective function:

$$f(\phi_{lo}) \leqslant f(\phi_j), \quad j = \overline{0, i}, \tag{15}$$

and also

$$f'(\phi_{lo})(\phi_{hi} - \phi_{lo}) < 0. \tag{16}$$

If $|\phi_{hi} - \phi_{lo}| < tol_\phi$ then the search is over, and ϕ_{lo} is returned.

A new iterate ϕ_{i+1} is defined as the unique local minimum of the *cubic polynomial* interpolating the values and the first derivatives of the objective function at the end points. As long as $f(\phi_{lo}) \leqslant f(\phi_{hi})$ and (16) is true, such a local minimum exists and is located between ϕ_{lo} and ϕ_{hi}:

$$\phi_{i+1} = \phi_{hi} - (\phi_{hi} - \phi_{lo}) \frac{f'(\phi_{hi}) + d_2 - d_1}{f(\phi_{hi}) - f(\phi_{lo}) + 2d_2},$$

where

$$\begin{aligned} d_1 &= f'(\phi_{lo}) + f'(\phi_{hi}) - 3\frac{f(\phi_{lo}) - f(\phi_{hi})}{\phi_{lo} - \phi_{hi}}, \\ d_2 &= \mathrm{sign}(\phi_{hi} - \phi_{lo})\sqrt{d_1^2 - f'(\phi_{lo})f'(\phi_{hi})}. \end{aligned}$$

If the new iterate ϕ_{i+1} is too close to ϕ_{lo}:

$$(\phi_{i+1} - \phi_{lo})/(\phi_{hi} - \phi_{lo}) < tol_{\phi_{lo}},$$

there is a potential danger for the iterations to fall into the stagnation loop; to avoid this, the iterate ϕ_{i+1} should be moved away from ϕ_{lo} al least as far as $tol_{\phi_{lo}}|\phi_{hi} - \phi_{lo}|$. We find that the value of $tol_{\phi_{lo}} = 0.05$ serves the purpose quite well.

If $|f'(\phi_{i+1})| \leqslant tol_{f'}$ then the search is over, and ϕ_{i+1} is returned.

The new iterate ϕ_{i+1} partitions the target interval in two subintervals. Which one we should pick to contunue the recursion?

If $\quad f(\phi_{lo}) < f(\phi_{i+1}) \quad$ then ϕ_{i+1} should replace ϕ_{hi} :

$$\phi_{hi} \leftarrow \phi_{i+1};$$

else if $\quad f'(\phi_{i+1})(\phi_{hi} - \phi_{lo}) < 0 \quad$ then ϕ_{i+1} should replace ϕ_{lo} :

$$\phi_{lo} \leftarrow \phi_{i+1};$$

otherwise we put

$$\begin{aligned} \phi_{hi} &\leftarrow \phi_{lo}, \\ \phi_{lo} &\leftarrow \phi_{i+1}. \end{aligned}$$

These rules guarantee that the new target interval satisfies both (15) and (16). Update the iterate counter $i \leftarrow i + 1$ and repeat the 3rd step.

In our numerical experiments we used $\Delta\phi = \pi/(2n)$ and $tol_{\phi} = 10^{-6}$. We deliberately set $tol_{f'} = 0$ ensure that all second order accurate reconstruction algorithms participating in the numerical experiments (see the next section on static tests) terminate under the same condition, i.e. whenever the polar angle of the interface normal stabilizes within tol_{ϕ}.

One should be careful choosing the *initial guess* for the optimization procedure: even if the objective function (7) has a unique *global* minimum, it may still possess some other *local* minima (see Figure 7, $\mu^* = 0.02$), which can distract the algorithm.

A safe initial guess ϕ_0 is provided by the polar angle of vector $\mathbf{x}_c(\Omega) - \mathbf{x}^*$. If Ω were a circle, this choice of ϕ_0 would give us the exact minimizer of (7). Even though this is not the case, the more isotropic the shape of the mixed cell is, the better the initial guess above approximates the objective function minimum.

We also claim that suggested ϕ_0 either specifies the point on the slope of a global minimum valley or marks the top of the hill that divides two global minima. Therefore the conservative value of $\Delta\phi$ guarantees to bring the search started at suggested ϕ_0 to a global minimum.

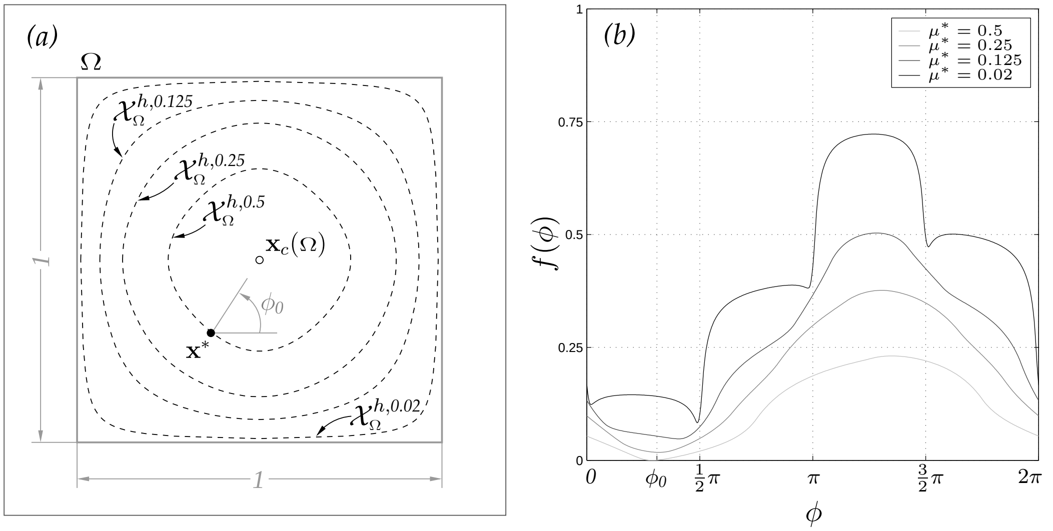


Figure 7. (a) The unit square cell Ω with a sample reference centroid $\mathbf{x}^*$ and sample centroid loci $\{\mathcal{X}_\Omega^{h,\mu^*}\}$ of the cutouts of various volume fractions ($\mu^* = 0.5, 0.25, 0.125, 0.02$).
(b) The respective objective function graphs. Each objective function has a unique global minimum; the one corresponding to $\mu^* = 0.02$ has three more local minima.

3 Static tests

The idea of static test for an interface reconstruction algorithm is pretty simple; given a particular shape of the reference component and a computational grid, one should:

1) calculate the moments of the reference component enclosed in each cell,
2) reconstruct the mixed cell interfaces based on the data obtained, and then
3) compare the resulting shape to the original one.

To calculate the moments of the reference component enclosed in a cell, one has to find the intersection of the cell with the original shape. If both the cell and the original shape are represented by convex polygons, it is possible to find the intersection in linear time [17, 16][2]; non-convex polygons for this purpose can always be decomposed into a set of convex polygonal parts.

[2] We used "chasing" algorithm [17] routine implemented by M. Staley in COnservative REmapper (CORE) library [24].

Along with MoF method we tested three VoF-PLIC methods for general polygonal grids:

- Least Square Gradient (LSG) with reciprocal quadratic distance weights[2], which we refer to as LSGQ,
- LVIRA [20, 19],
- Swartz algorithm [28, 12].

The choice of the last two methods is pretty obvious: these are the only original second order accurate VoF-PLIC methods suitable for unstructured grids. Both LVIRA and Swartz algorithms are *iterative* and rely on a *reasonable initial guess*, usually provided by a *direct* VoF-PLIC method (LSGQ in our case). More details on implementation of these VoF-PLIC algorithms can be found in Appendix.

The first order accurate algorithms are represented by LSG; it requires only a list of neighbors for each mixed cell. Frankly, the only real alternative to LSG is Green-Gauss algorithm [2], but the latter requires adjacent cells to be properly ordered. On a uniform rectangular grid both LSGQ and Green-Gauss give *the same* normal estimate as the Youngs algorithm [31].

The results produced by these PLIC algorithms for six different shapes are presented on Figures 8 and 9. Although original shapes are not shown, their names are self-explanatory. As one can see, MoF method has higher resolution than VoF alternatives, adds no blurring, and takes the maximum out of PLIC approximation: polygonal shapes "zigzags" and "compass rose", which are specially designed to have a piecewise-linear boundary that can be captured on the 16×16 grid with a single linear segment per cell, are reproduced by MoF algorithm exactly.

Error quantification. Our next step is quantification of the interface reconstruction error. With focus of discussion shifting from a single mixed cell to the entire computational grid, it is necessarily to change the notation. Let $\Omega \subset \mathbb{R}^2$ now represent a polygonal domain *quasi-uniformly* partitioned into N polygonal cells $\{\Omega_i\}_{i=1}^N$:

$$\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \; \forall \; i \neq j;$$

open $\omega^* \subset \Omega$ and $\omega_h^* \subset \Omega$ represent the original and the reconstructed shapes of the reference component. The intersections of ω^* and ω_h^* with a single cell $\Omega_i, \; i = 1, N$ will be referred to as ω_i^* and $\omega_{h,i}^*$ respectively.

Several ways to quantify the interface reconstruction error were mentioned in Subsection 2.1. Since we already had in our possession a polygon intersec-

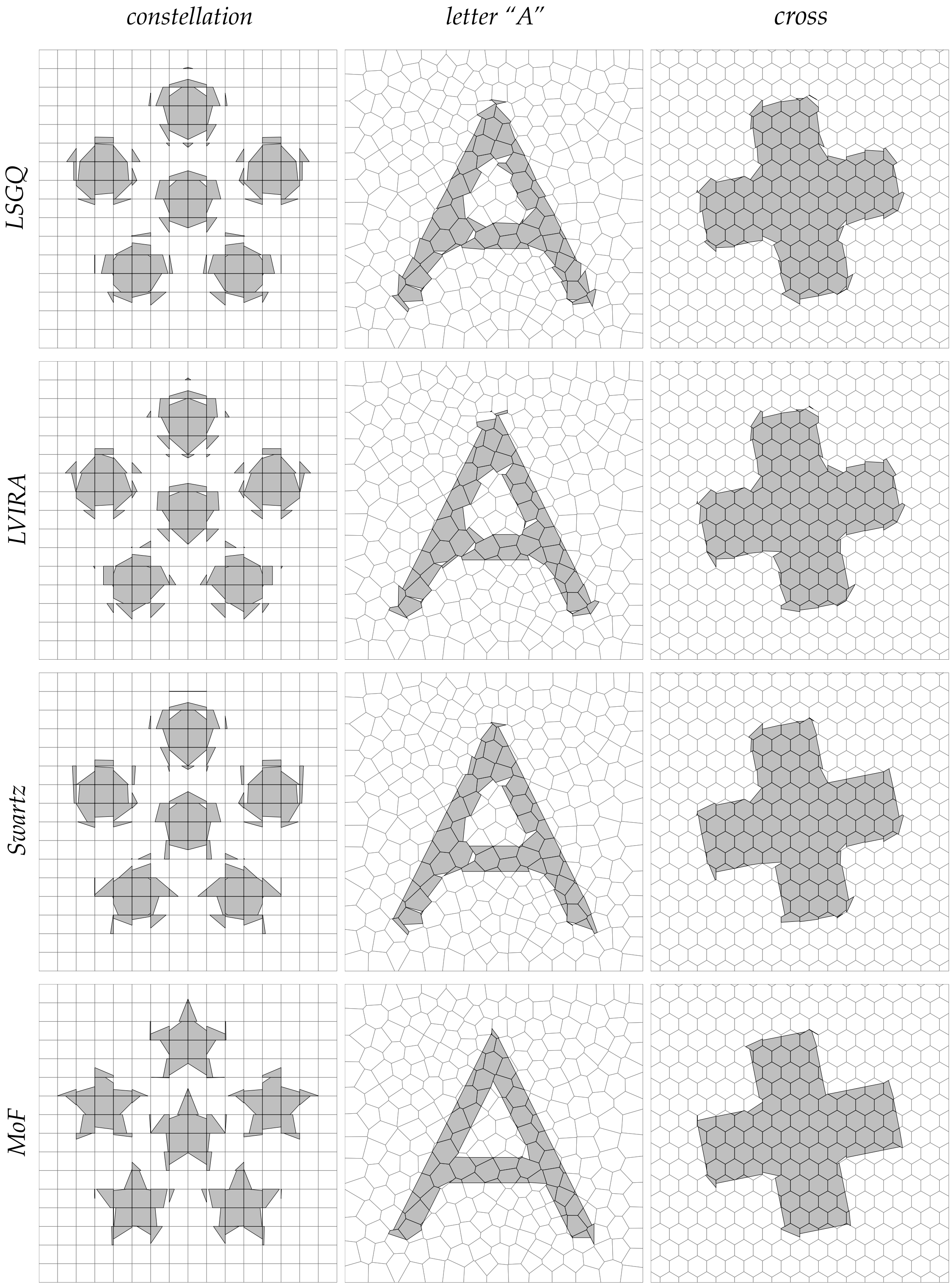


Figure 8. Results of the interface reconstruction with various PLIC techniques.

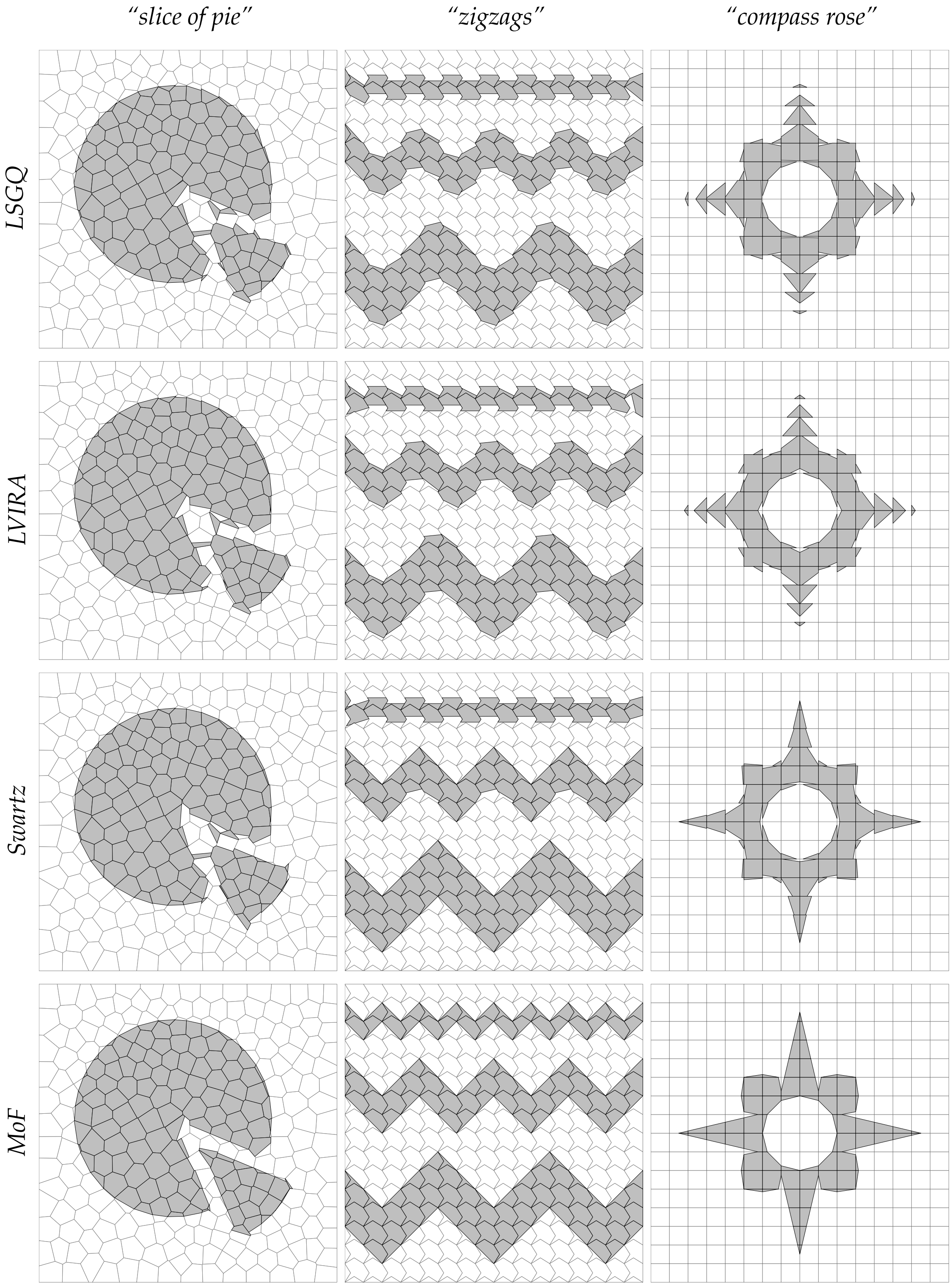


Figure 9. More interface reconstruction results.

tion routine[3)], we decided to measure the *area of the symmetric difference between ω^* and ω_h^**:

$$\Delta\omega = |\,\omega^* \triangle\, \omega_h^*| = \sum_{i=1}^{N} |\,\omega_{h,i}^* \triangle\, \omega_i^*|.$$

The interface reconstruction algorithms we examine are volume-conservative, i.e. $|\omega_{h,i}^*| = |\omega_i^*|,\ i = \overline{1,N}$, therefore

$$|\,\omega_{h,i}^* \triangle\, \omega_i^*| = \begin{cases} 2\big(|\omega_i^*| - |\omega_i^* \cap \omega_{h,i}^*|\big), & \text{if } \Omega_i \text{ is a mixed cell,} \\ 0, & \text{otherwise,} \end{cases}$$

which yields the following practical expression for $\Delta\omega$:

$$\Delta\omega = 2 \sum_{i=1}^{M} \big(|\omega_i^*| - |\,\omega_i^* \cap \omega_{h,i}^*|\big); \tag{17}$$

here M is the total number of mixed cells, without loss of generality we can assume that all mixed cells are enumerated from 1 to M.

This area, divided by the length of the original interface $|\partial\omega^*|$, gives the *average distance between the reconstructed and original interfaces*. To make this quantity scale-independent, we normalize it by a *characteristic size $\tilde{L}$* of the original shape ω^*; the result is referred to as a *normalized average deviation*:

$$\tilde{\varepsilon}_{avg} = \frac{1}{\tilde{L}} \frac{|\,\omega^* \triangle\, \omega_h^*|}{|\partial\omega^*|}.$$

Another important error characteristics we can get with $\{|\,\omega_i^* \cap \omega_{h,i}^*|\}_{i=1}^{M}$ is the *maximum of the normalized average deviation per (mixed) cell*:

$$\tilde{\varepsilon}_{max} = \max_{i=\overline{1,M}} \frac{1}{\tilde{L}} \frac{|\,\omega_i^* \triangle\, \omega_{h,i}^*|}{|\Gamma(\omega_{h,i}^*)|},$$

which indicates the worst local error attainable.

Direct error mesurements. To figure out how these errors scale with mesh refinement we conducted a series of direct error measurements on a sequence of uniform rectangular grids ($\Omega\ =]0,1[\times]0,1[$) with mesh spacing h varying from 1/2 to $1/2^{10}$ ($h_k = 1/2^k,\ k = \overline{1,10}$).

Each PLIC algorithm was tested against two shapes:

1) circle of radius $\tilde{L} = 0.25$ centered at $(0.5 + 1/17, 0.5 + 1/41)$;

[3)] Due to COnservative REmapper (CORE) library [24] by M. Staley.

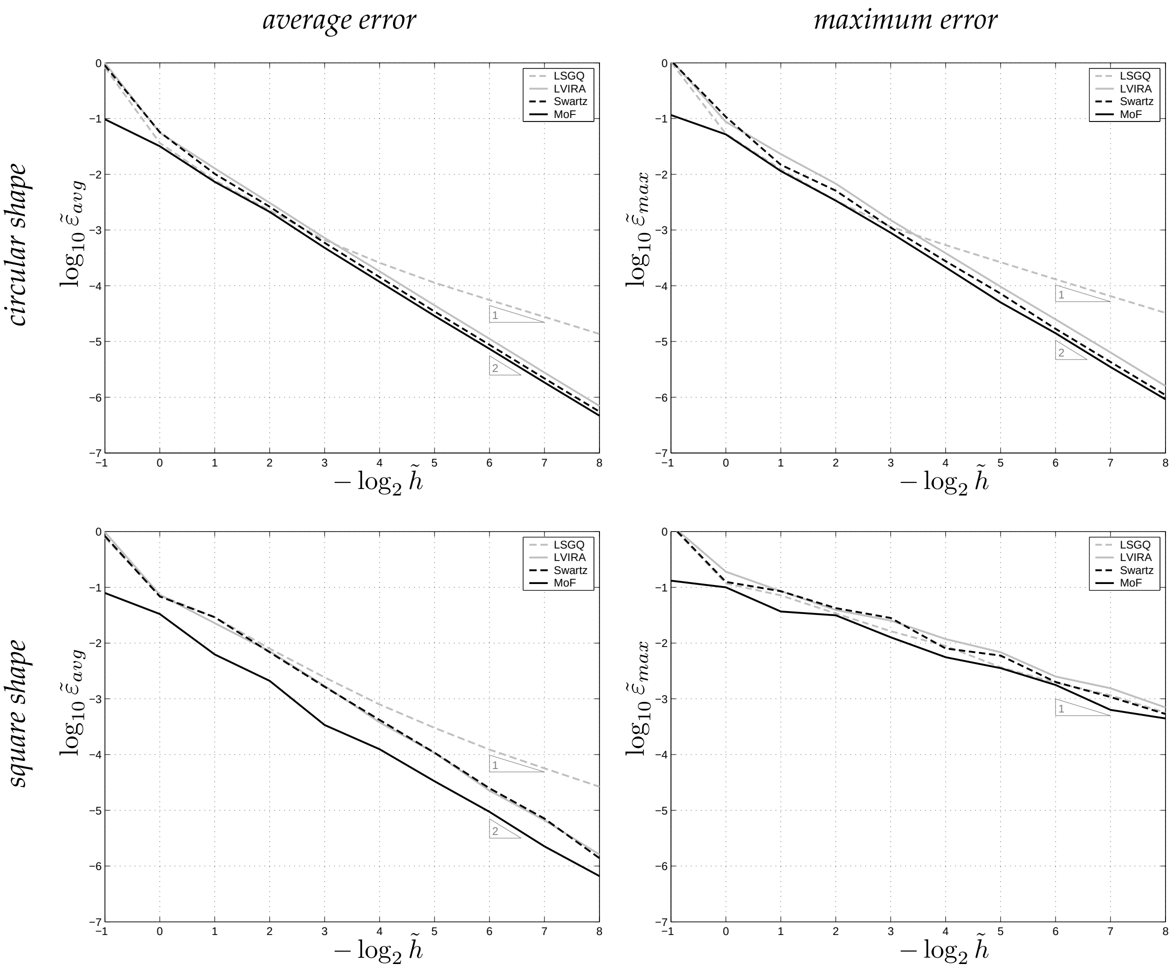


Figure 10. Interface reconstruction error.

2) $2\tilde{L} \times 2\tilde{L} = 0.5 \times 0.5$ square centered at $(0.5 + 1/17, 0.5 + 1/41)$ and rotated counter-clockwise by $\pi/16$ radians;

Both shapes have the same characteristic scale $\tilde{L} = 0.25$, their location and orientation were chosen to eliminate the influence of the symmetry factor on error readings.

The results of these tests are presented on the graphs (Figure 10), where error is related to the normalized mesh spacing $\tilde{h} = h/\tilde{L}$. Both horizontal and vertical axes are logarithmic.

Let us first comment the circle test results. As one might have anticipated, LVIRA, Swartz and MoF algorithms provide second order accurate approximation to the C^2-smooth interface, while LSGQ is only first order accurate. Uniform curvature of the interface explains similar behavior of both average and maximum errors, although average error is somewhat lower and less volatile due to the aggregate nature of this indicator. MoF algorithm results in the lowest error. Average errors of LVIRA and Swartz algorithms are $50+\%$ and $18+\%$ higher than respective MoF error. Even though LSGQ is asymptotically less accurate than LVIRA and Swartz, the former algorithm exhibits significantly smaller error (just 6% above MoF) on coarse and medium scales and starts to loose the advantage over LVIRA and Swartz only after the cell size drops below 1/8 of the interface curvature radius.

In the second test the original interface is not smooth, and that is why all four methods demonstrate maximum local error of order $O(h)$. Asymptotic average error of LSGQ is also $O(h)$, while for all other three methods this indicator is only $O(h^2)$. This behavior has a clear explanation: all second order accurate methods are able to capture straight segments of the interface exactly, and only in the mixed cells surrounding sharp corners of the original polygonal shape the error is not zero. With $O(1)$ mixed cells contributing to the average error, global $\Delta\omega$ is only $O(h^2)$ (see (4)), which results in $\tilde{\varepsilon}_{avg} = \Delta\omega/(8\tilde{L}^2) = O(h^2)$ average deviation. Once again MoF algorithm demonstrates the highest accuracy. Since there is only one mixed cell per vertex of polygonal interface that contributes to MoF average error against three to four for competitors, MoF average error is at least two times lower than the respective average error of any other method. We would also like to emphasize the superiority of MoF algorithm over the VoF-PLIC methods on a very coarse scale.

CPU time mesurments. In terms of CPU time the clear winner is LSGQ, which is about five times faster than MoF and Swartz methods and seven times faster than LVIRA. Common software components shared between the implementations of different methods allow us to claim a non-discriminative character of these results.

4 Advection scheme example

In order to employ MoF interface reconstruction algorithm in a fluid flow simulation, we have to equip it with appropriate advection scheme. Below we present an example of the transport algorithms that can be used to update the volume and centroid data in *incompressible* fluid flow simulations. The scheme can be characterized as a *Lagrangian remap*.

The *solenoidal* velocity field $\mathbf{v}(\mathbf{x}, t)$ is assumed to be given analytically, therefore no explicit constraint on the Courant number $CFL = v\,\Delta t/h$ is imposed (here Δt is the time step, h is the local mesh spacing, and v is the local flow speed).

4.1 Volume transport

Given the location of mixed cell interfaces at $t_{k-1} = \Delta t\ (k-1)$, one can evaluate the content of cells at $t_k = t_{k-1} + \Delta t$ as follows:

```
**for each** cell $\Omega_i,\ i = \overline{1, N}$ **do**
    track $\Omega_i$ *back in time* to $t = t_{k-1}$ to identify the *Lagrangian prototype* $\Omega_{i,k-1}$ of the cell;
    find the volume $\tilde{m}_{i,k-1}$ of the reference phase enclosed in Lagrangian prototype $\Omega_{i,k-1}$;
    put the volume $m_{i,k}$ of the reference component in $\Omega_i$ equal to $\tilde{m}_{i,k-1}$.
**end do**
```

The vertices of polygonal cell Ω_i are tracked back along the streamlines by means of the 4-th order Runge-Kutta scheme and then connected in proper order by the straight segments. This results in a polygon that we consider to be a *discrete Lagrangian prototype* $\Omega_{i,k-1}$ of cell Ω_i (see Figure 11).

Using a polygon intersection routine[4], we find the intersections of $\Omega_{i,k-1}$ with all cell fractions at $t = t_{k-1}$ covered by the prototype. One can significantly accelerate the search of such cell fractions by partitioning the bounding box of the entire computational domain into 2D array of rectangular bins and presorting all the mesh cells among these bins based on the centroid location. If $CFL \leqslant 1$, it is sufficient to intersect the prototype only with the cell fractions of Ω_i and its direct neighbors. Moreover, if $CFL \leqslant 1$ and Ω_i along with all its neighbors is empty at $t = t_{k-1}$, then it is guaranteed to stay empty the next discrete moment of time, and there is no need to perform any polygon intersections in this case at all.

In a *linear* velocity field, which is known to preserve straight lines, the *true Lagrangian prototype* of a polygonal cell is always a polygon, and the accuracy of the remapping is limited only by the accuracy the integration scheme. For a p-th order accurate scheme the *area defect* $|\Omega_{i,k-1}| - |\Omega_i|$ is estimated as $O(h\,\Delta t^{p+1})$.

In a nonlinear velocity field straight lines are not preserved, and the Lagrangian prototype of a polygonal cell is not exactly a polygon. Therefore by ignoring the curvature of the prototype edges we introduce additional $O(h^3 \Delta t)$ area defect.

Even a small area defect can eventually cause the algorithm to halt. Indeed, if the volume of the reference component $\tilde{m}_{i,k-1}$ enclosed in the prototype $\Omega_{i,k-1}$ exceeds the capacity of the cell $|\Omega_i|$, we are in trouble. Another, less critical, situation occurs when the prototype, being filled with the reference component, happens to have the area $|\Omega_{i,k-1}|$ smaller than $|\Omega_i|$. In this case the cell becomes mixed, even though its prototype contains only one component.

[4] Due to COnservative REmapper (CORE) library [24] by M.Staley.

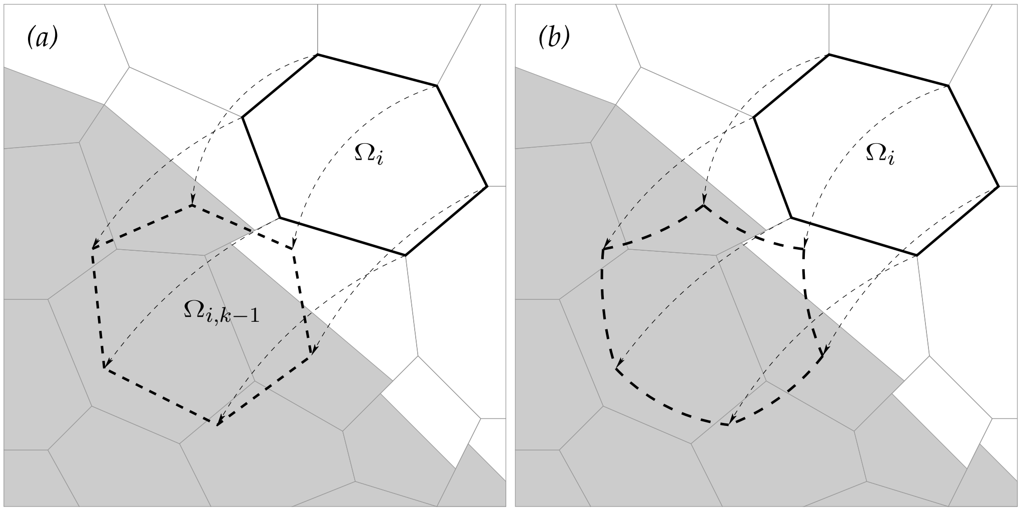


Figure 11. The discrete (a) and true (b) Lagrangian prototypes of the cell.

In order to fix these flaws we use a post-remapping *repair* procedure. For every cell $\Omega_i,\ i = \overline{1,N}$ we specify the lower $\underline{m}_{i,k}$ and the upper $\overline{m}_{i,k}$ bounds of the reference fraction volume $m_{i,k}$ allowed:

$$\begin{array}{lll} \underline{m}_{i,k} = & \overline{m}_{i,k} = 0 & \text{if the prototype is empty,} \\ \underline{m}_{i,k} = & \overline{m}_{i,k} = |\Omega_i| & \text{if the prototype is full,} \\ \underline{m}_{i,k} = 0, & \overline{m}_{i,k} = |\Omega_i| & \text{otherwise,} \end{array} \tag{18}$$

and then force each volume $m_{i,k},\ i = \overline{1,N}$ to fit into these bounds:

```
for each cell $\Omega_i,\ i = \overline{1,N}$ do
    if $\Omega_i$ is overfilled ($\overline{m}_{i,k} < m_{i,k}$) then
        try to redistribute the excess between the non-overfilled neighbours
        while there is still some excess remained do
            redistribute them among the next layer of the surrounding cells
        end do
    else if $\Omega_i$ is underfilled ($m_{i,k} < \underline{m}_{i,k}$) then
        try to compensate the shortfall by borrowing from the non-underfilled neighbours
        while there is still some shortage of material do
            borrow it from the next layer of the surrounding cells
        end do
    end if
end do
```

Due to the local nature of the area defect, the redistribution usually involves only direct neighbours of the cell. Therefore the complexity of the whole repair step comes to the total of $O(N)$.

4.2 Centroid transport

The technique above is easily extensible for advancing centroid positions in time. Whenever a non-empty intersection of discrete Lagrangian prototype $\Omega_{i,k-1}$ with any cell fraction is detected, one has to calculate not only the area but also the first moment of the intersection. After all the intersections are found, the centroid of the reference fraction of $\Omega_{i,k-1}$ is calculated as the moment to volume ratio and is tracked forth along the streamlines to determine the position of the reference centroid in cell Ω_i. There is no need to track the centroid if the prototype contains only one component: with (18) bounds the repair step guarantees that the cell will stay pure as well.

Whenever the velocity field is linear in space, the actual centroid velocity of any volume of fluid coincides with the field velocity at the centroid location:

$$\frac{d}{dt}\mathbf{x}_c(\omega) = \mathbf{v}(\mathbf{x}_c(\omega)). \tag{19}$$

Therefore, in this case, the trajectory of real centroid follows a streamline and the reference centroid error is due only to approximate integration ($O(\Delta t^{p+1})$ for a p-th order scheme).

For a nonlinear velocity field the identity (19) is void. As long as $\mathbf{v}(\mathbf{x}, t)$ is twice differentiable in $\mathbf{x}$, the following estimate, given by Taylor expansion, holds:

$$\frac{d}{dt}\mathbf{x}_c(\omega) = \mathbf{v}(\mathbf{x}_c(\omega)) + O(d^2). \tag{20}$$

Here $d = \text{diam}(\omega)$ is the diameter of the volume advected. The cumulative reference centroid error in this case is $O(d^2\Delta t)$.

Since the reference centroid error is determined by the diameter of the advected volume, some accuracy improvement can be gained by advecting the fluid in smaller pieces. The reference fraction of Lagrangian prototype is originally assembled from elementary intersections of $\Omega_{i,k-1}$ with the reference fractions of cells on the previous time step. Therefore, instead of advecting the aggregated centroid, one may choose to advect centroids of these elementary parts separately and only then proceed with their aggregation (Figure 12a).

Also, whenever the volume of the complementary phase in the prototype is smaller than the volume of the reference component ($|\Omega_{i,k-1} \backslash \overline{\omega}_{i,k-1}| \lessapprox |\omega_{i,k-1}|$),

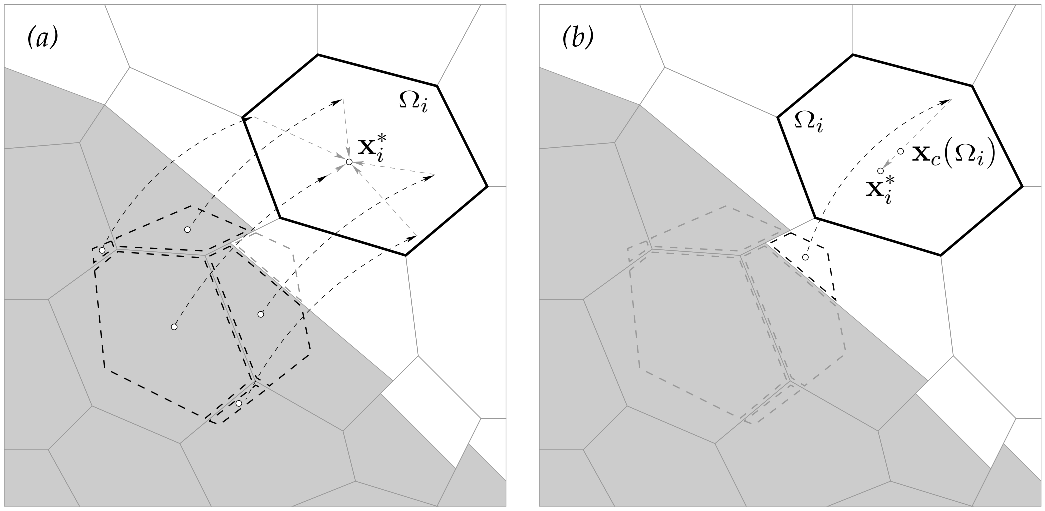


Figure 12. Two ways to improve the reference centroid accuracy: (a) by advecting the reference fraction in pieces, (b) by advecting a smaller complementary fraction.

one should advect the complementary centroid instead and then use the identity

$$\mathbf{x}_c(\omega_i)\,|\omega_i| + \mathbf{x}_c(\Omega_i \backslash \overline{\omega}_i)\,|\Omega_i \backslash \overline{\omega}_i| = \mathbf{x}_c(\Omega_i)\,|\Omega_i|$$

to get the position of the reference centroid (Figure 12b).

4.3 Numerical examples

Example 1: Solid Rotation of Letter A. The first set of snapshots (Figure 13) is due to advection of "A" glyph in the solid rotation field:

$$\mathbf{v}((x,y),t) = \begin{bmatrix} -(y-y_0) \\ +(x-x_0) \end{bmatrix}, \quad (x_0,\ y_0) = (0.5,\ 0.5).$$

Other parameters of the numerical experiment:

computational domain	$]0,\ 1[\times]0,\ 1[$,
simulation time	$T = 2\pi$,
computational grid	1024 polygonal cells,
number of time steps	$N_T = 144$.

The velocity field is linear so that the error introduced by interface reconstruction prevails over the advection error. All VoF-PLIC methods oversmooth the interface, while MoF method demonstrates a good ability to preserve the shape of the glyph.

Example 2: Reversible vortex. The second test case [22] simulates the evolution of the round "blot"

$$\{\, \mathbf{x} \in \mathbb{R}^2 \;\mid\; ||\,\mathbf{x} - \mathbf{x}_0|| \leqslant R \,\}, \quad \mathbf{x}_0 = (0.5,\ 0.75),\ R = 0.15$$

in the vortex field:

$$\mathbf{v}((x,y),t) = \left[\begin{array}{c} +\sin^2(\pi x)\sin(2\pi y) \\ -\sin^2(\pi y)\sin(2\pi x) \end{array}\right] \cos(\pi t/T).$$

The cosine multiplier gradually "decreases the power" of the vortex until the complete stop ($\mathbf{v}(\mathbf{x},t) \equiv \mathbf{0}$) at $t = T/2$, and then starts to "power up" the reversed vortex. Since

$$\mathbf{v}(\mathbf{x},t) = -\mathbf{v}(\mathbf{x}, T-t), \quad 0 < t < T,$$

the exact integration of the fluid motion from $t = 0$ to $t = T$ should result in zero changes. Therefore by comparing the final configuration against the initial setup, one can get the idea of accuracy of numerical method employed. For easier judgment we overlayed the snapshots (Figure 14) with the interface line obtained with tracking technique [1].

Other parameters of the experiment:

computational domain	$]0,\ 1[\times]0,\ 1[$,
simulation time	$T = 8$,
computational grid	uniform, 32×32 cells,
number of time steps	$N_T = 256$.

Although the final shape obtained with MoF method demonstrates better overlap with the initial shape than the VoF-PLIC alternatives, the latter result in much smoother interface. The reason for this is that VoF methods, due to the aggregate nature of the interface normal estimate, have a tendency to oversmooth the interface, which on the dynamic level can be interpreted as an artificial numerical surface tension. This feature becomes determinative when a VoF method is forced to work on the edge of the resolution limit: by the moment of maximum stretch ($t = T/2$) the shape of the reference component becomes so thin that it breaks into a series of separate blobs. MoF method does not exhibit this kind of behavior and gives much better approximation of real interface at $t = T/2$.

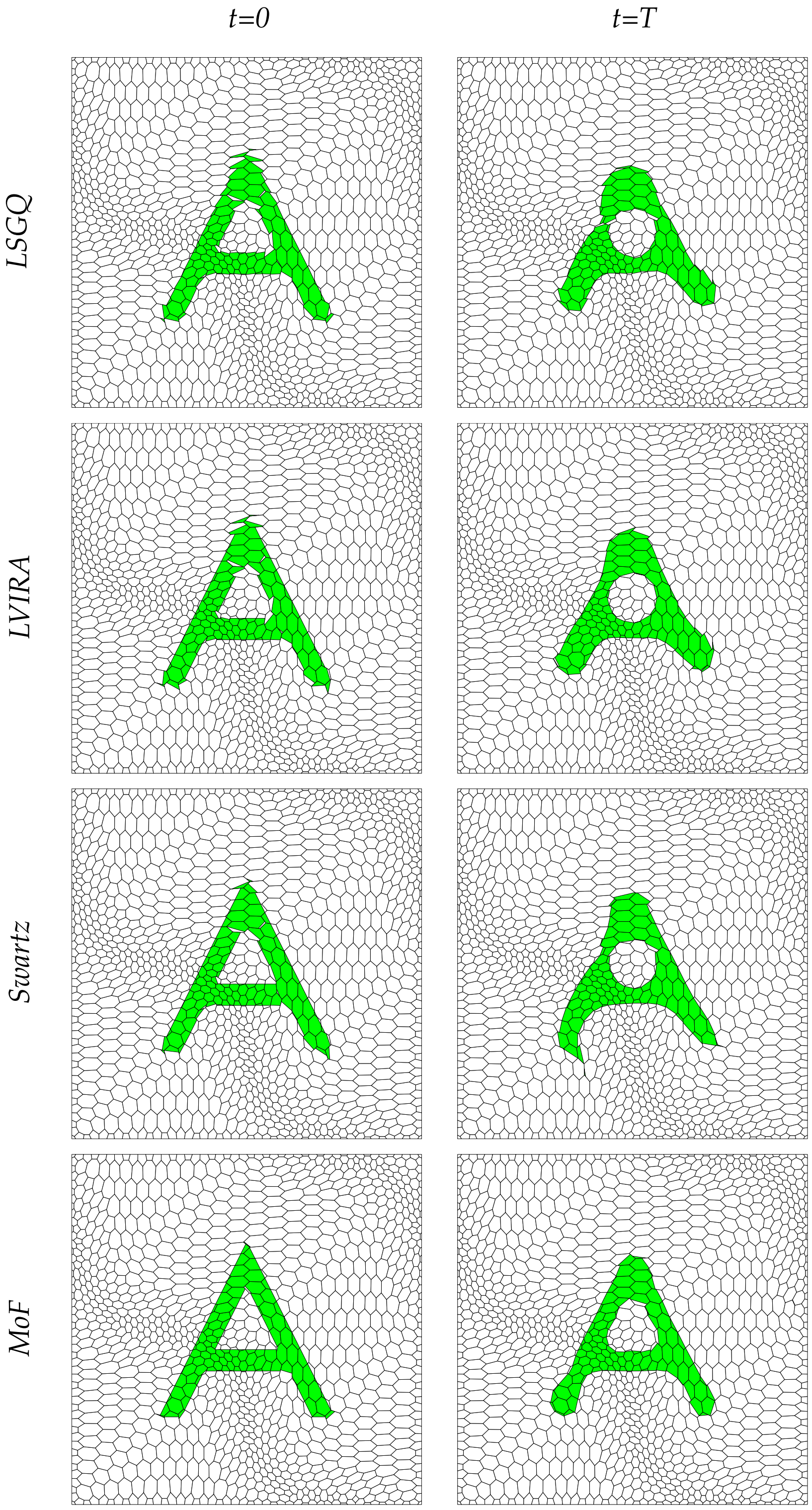


Figure 13. Solid rotation test results.

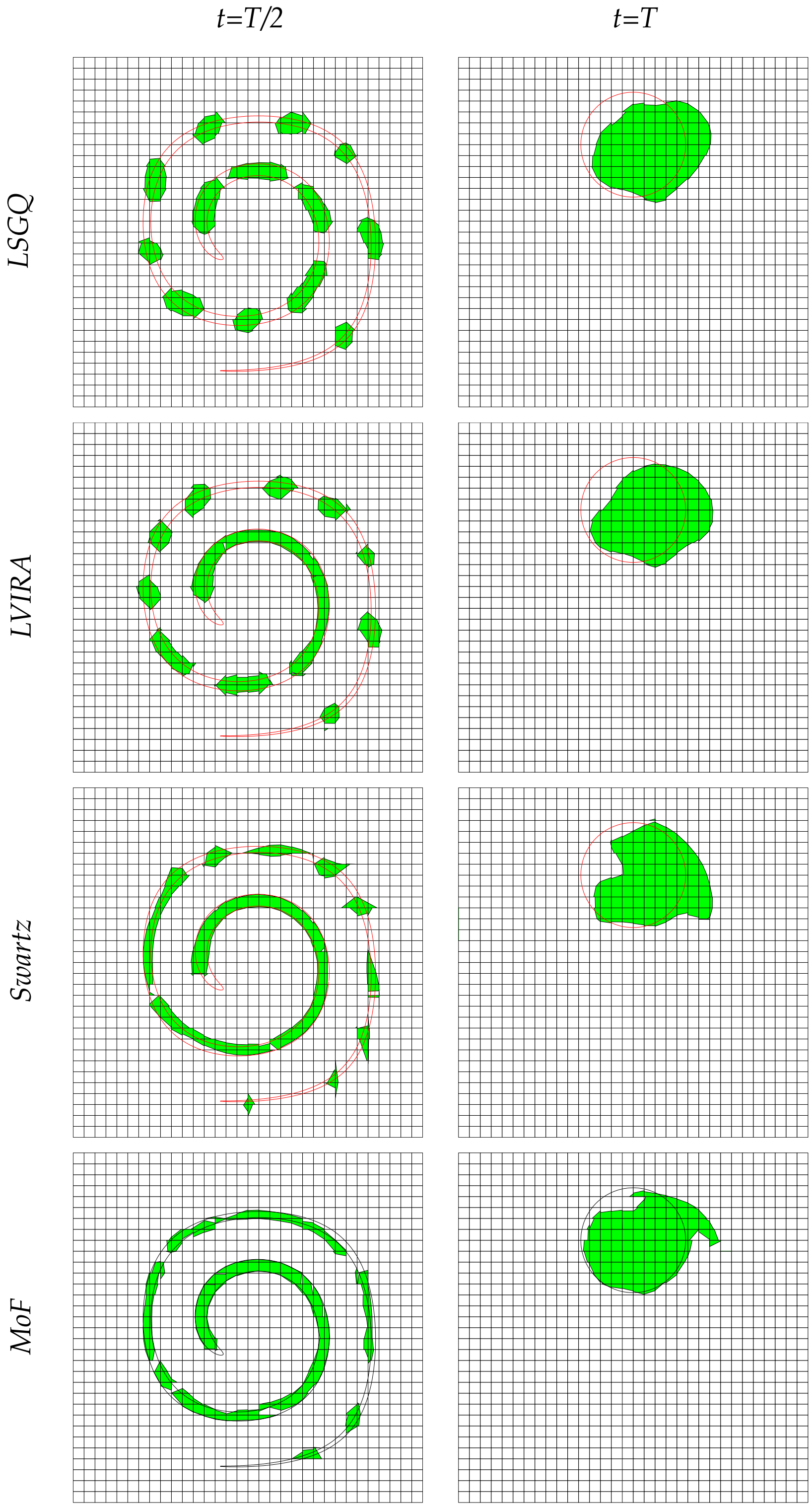


Figure 14. Reversible vortex test results.

5 Acknowledgment

Authors would like to thank Blair Swartz, Rao Garimella, Konstantin Lipnikov, James Mac Hyman, Martin Staley, William Rider, Robert Lowrie, Douglas Kothe, Marianne Francois, Stewart Mosso, Tariq Aslam, and David Benson for useful discussions and valuable comments.

Authors also acknowledge the support of the Advanced Simulation and Computing (ASC) program at the Los Alamos National Laboratory.

Appendix

A Implementation details of the VoF-PLIC algorithms

In order to evaluate the interface normal, all VoF-PLIC algorithms require the list of cells adjacent to a given mixed cell. We consider two cells to be adjacent if they share either an edge or a vertex.

LSGQ and LVIRA. Implementations of LSGQ and LVIRA follow exactly the descriptions given in [2] and [19] respectively.

The initial guess for LVIRA is given by LSGQ output. The minimum of LVIRA objective function is considered to be localized when the polar angle of the interface outward normal iterate stabilizes within $tol_\phi = 10^{-6}$ radians.

Swartz algorithm. We believe that Swartz algorithm is somewhat underrepresented in literature and therefore would like to discuss it in more detail.

The idea of the method is based on the fact that for a pair of adjacent mixed cells there exists a common linear interface that satisfies arbitrary given volume fractions. In the context of structured rectangular grid the common linear interface can be found analytically [6, 10]. For general polygonal grids B. Swartz has introduced an iterative procedure [28] that converges quadratically to a common linear interface:

Given an initial VoF-PLIC interface approximation in both mixed cells,

1) connect the medians of interfaces with a straight segment;

2) use the normal of this segment as a new iterate for the normal of common linear interface;

3) build a new linear interface in each cell using this normal;

4) repeat these three steps until the direction of the common interface normal establishes within some small tolerance (we used $tol_\phi = 10^{-6}$).

Our implementation defines the direction of the interface normal in each mixed cell as an arithmetic average of common interface normals due to all possible mixed neighbours. This strategy is somewhat different from the scheme described by S. Mosso and B. Swartz in [12] and is actually the reason why the algorithm we implemented is called *Swartz*, not *Mosso-Swartz*.

In addition to the procedure above, further referred to as *internal iterations*, Mosso-Swartz algorithm specifies an *external iterative process* that loops through the list of mixed cell, updating the interfaces one in a time in *Gauss-Seidel manner*, i.e. as soon as a new interface is evaluated. By means of numerical experiment we figured out that such external iterations result in excessive numerical surface tension, oversmooth the interface and decrease the resolution of the algorithm.

Therefore we do not use external iterations at all and update the interfaces just one time, in *Jacobi manner*, i.e. only after all the new interfaces have been evaluated from the original initial guess.

Whoever wants to implement the internal iterations should know that *a common linear interface satisfying given volume fractions is not unique*. The iterative process, depending on initial guess and details of implementation, may converge to either of them. As it has been mentioned before, the initial guess for internal iterations in our implementation is provided by LSGQ algorithm.

One should also understand that the segment connecting the medians of the adjacent interfaces allows *two choices of the outward normal.* Between the two opposite normals we choose the one that has a positive projection on the previous normal iterate.

We do not perform the internal iterations for a pair of adjacent mixed cells at all, unless the initial outward normals differ by less than 45 degrees. The rationale of this *discrimination strategy* comes clear if we start to look at Swartz iterations as at the second order accurate "upgrade" of initial interface approximation: if two initial normals already differ significantly, then the common linear interface is highly unlikely to give an adequate approximation to the original one, and therefore should not contribute to the average. This way the "upgrade" kicks in only for those pairs of adjacent cells that carry information about the same smooth segment of the original interface.

References

[1] E. Aulisa, S. Manservisi, and R. Scardovelli. A surface marker algorithm coupled to an area-preserving marker redistribution method for three-dimensional interface tracking. *Journal of Computational Physics*, 197(2):555–584, Jul 2004.

[2] Timothy J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In D. Kröner, M Ohlberger, and C. Rohde, editors, *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws. Proceedings of the International School on Theory and Numerics for Conservation Laws, Freiburg/Littenweiler, October 20–24, 1997*, Lecture Notes in Computational Science and Engineering, pages 271–273. Springer-Verlag, Berlin Heidelberg, 1999.

[3] David J. Benson. Eulerian finite element methods for micromechanics of heterogeneous materials: Dynamic prioritization of material interfaces. *Computer Methods in Applied Mechanics and Engineering*, 151(3–4):343–360, Jan 1998.

[4] David J. Benson. Volume of fluid interface reconstruction methods for multi-material problems. *Applied Mechanics Reviews*, 55(2):151–165, Mar 2002.

[5] J. Frédéric Bonnans, J. Charles Gilbert, Claude Lemaréchal, and Claudia A. Sagastizábal. *Numerical Optimization. Theoretical and Practical Aspects*. Springer-Verlag, Berlin Heidelberg, 2003.

[6] R. DeBar. Fundamentals of the KRAKEN code. Technical Report UCID-17366, Lawrence Livermore National Laboratory, Livermore, CA, 1974.

[7] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183:83–116, 2002.

[8] J. Glimm, J. W. Grove, X. L. Li, K. M. Shyue, Y. N. Zeng, and Q. Zhang. Three-dimensional front tracking. *SIAM Journal on Scientific Computing*, 19(3):703–27, May 1998.

[9] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–25, Jan 1981.

[10] W. H. McMaster. Computer codes for fluid-structure interactions. Technical Report UCRL-89724, Lawrence Livermore National Laboratory, Livermore, CA, 1984.

[11] J. P. Morris. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 33:333–353, Jun 2000.

[12] S. J. Mosso, B. K. Swartz, D. B. Kothe, and R. C. Ferrell. A parallel, volume-tracking algorithm for unstructured meshes. In P. Schiano, A. Ecer, J. Periaux, and N. Satofuka, editors, *Parallel Computational Fluid Dynamics: Algorithms and Results Using Advanced Computers*, pages 368–375. Elsevier Science, 1997.

[13] Stewart Mosso and Sean Clancy. A geometrically derived priority system for Young's interface reconstruction. Technical Report LA-CP-95-0081, Los Alamos National Laboratory, Los Alamos, NM, 1995.

[14] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[15] W. F. Noh and P. Woodward. SLIC (simple line interface calculation). In A. I. van der Vooren and P. J. Zandbergen, editors, *5th International Conference on Numerical Methods in Fluid Dynamics*, pages 330–340. Springer-Verlag, 1976.

[16] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, second edition, 1988.

[17] J. O'Rourke, C.-B. Chien, T. Olson, and D. Naddor. A new linear algorithm for intersecting convex polygons. *Compute. Graph. Image Process.*, 19:384–391, 1982.

[18] S. Osher and R. P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169(2):463–502, May 2001.

[19] J. E. Pilliod and E. G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199(2):465–502, Sep 2004.

[20] E. G. Puckett. A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction. In H. Dwyer, editor, *Proceedings of the 4th International Symposium on Computational Fluid Dynamics*, pages 933–938, Davis, CA, 1991.

[21] E. G. Puckett and J. S. Saltzman. A 3D adaptive mesh refinement algorithm for multimaterial gas dynamics. *Physica D*, 60(1–4):84–93, Nov 1992.

[22] William J. Rider and Douglas B. Kothe. Reconstructing volume tracking. *Journal of Computational Physics*, 121(2):112–152, Apr 1998.

[23] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31:567–603, 1999.

[24] Martin Staley. CORE: Conservative remapper. Technical Report LA-UR-04-8104, Los Alamos National Laboratory, Los Alamos, NM, 2004.

[25] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. An adaptive level set approach for incompressible two-phase flows. *Journal of Computational Physics*, 148:81–124, 1999.

[26] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. Improved level set method for incompressible two-phase flows. *Computers & Fluids*, 27(5–6):663–680, Jun–Jul 1998.

[27] M. Sussman and E. G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301–337, Aug 2000.

[28] Blair Swartz. The second-order sharpening of blurred smooth borders. *Mathematics of Computation*, 52(186):675–714, Apr 1989.

[29] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, and Y. J. Nas, S. amd Jan. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169(2):708–759, May 2001.

[30] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1):25–37, May 1992.

[31] D. L. Youngs. An interface tracking method for a 3D Eulerian hydrodynamics code. Technical Report AWRE/44/92/35, Atomic Weapon Research Establishment, Aldermaston, Berkshire, UK, Apr 1987.